SOCIETY OF ACTUARIES Forecasting and Futurism Section

ISSUE 10 | DECEMBER 2014

Forecasting & Futurism

NEWSLETTER

Predictably Irrational, by Dan Ariely The Hidden Forces that Shape our Decisions

By Ben Wolzenski

ISSUE 10 | DECEMBER 2014

Forecasting & Futurism

NEWSLETTER

Published by the Forecasting and Futurism Section Council of the Society of Actuaries

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 Forecasting and Futurism Section Web page at *www.soa.org/forecasting-futurism*.

FROM THE EDITOR:

精算师 **Master of** Accurate Calculations … Really?

By Dave Snell

A Chinese word for Actuary (精算师 Jīngsuàn shī) is very flattering. It can be translated as master, or teacher, of accurate calculations. That is far nicer than the English counterpart, which might be confused with a place to store dead actors. The implication is that we are the experts when it comes to accuracy. We are the rock stars of all things mathematical.

Alas, the actuarial profession is being assailed on several fronts by other professionals with excellent mathematical and business skill sets; and we have to prove our continued worth as masters of the calculations of risk management.

I sometimes feel a bit like Alice, in chapter two of Through the Looking Glass, by Lewis Carroll:

> "Well, in our country," said Alice, still panting a little, "you'd generally get to somewhere else—if you run very fast for a long time, as we've been doing."
>
> "A slow sort of country!" said the Queen. "Now, here, you see, it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!"

Our world of risk management is moving very fast. The tools and skill sets we learned in previous years may not be adequate for the complexities of market risks, country risks, credit or default risks, systemic risks, political risks, foreign exchange risks, interest rate risks, reputational risks, social media risks, pandemic risks, organizational and operational risks, innovation risks, employee risks, compliance risks, behavioral risks and on and on …

In this new world, we must not expect that a career will remain the same for decades. Our incoming chair of the Forecasting & Futurism Section, Doug Norris, describes the new paradigm in his article "Five Years is a Lifetime (Personal Forecasting)." Doug chronicles how dramatically his own career has changed in the past five years, and how the Forecasting & Futurism Section has become an increasingly important part of it. His article leads up to a thought provoking sequel to the interview question of "where do you want to

be in five years," with "what can you do today to help you get there?"

When I first entered the actuarial profession, we made '"best estimate" assumptions about interest, mortality, and expenses. That three-factor approach was pretty much the universe of risks for most actuaries. Now, best estimate approaches alone seem naïve and we wrap stochastic runs around them. We build sophisticated models and then lament that our models based strictly on logic must also consider the illogical—behavioral economics has emerged as an important discipline in the risk management world.

In this issue, Ben Wolzenski gives us a short course in behavioral economics through his review of *Predictably Irrational,* a best seller written by Dan Ariely. Dan is a professor of both psychology and behavioral economics. His experiments and resultant insights suggest that in order to

CONTINUED ON **PAGE 4**

remain "masters of accurate calculations" we have to better understand which calculations are relevant and which ones are naïve, based only on logic, not on the reality of some irrational behavior.

Sometimes we get so focused on the complex, that we overlook the simple. Kurt Wrobel makes that point very well in his article "Risk Management and the Power of Simplicity," which we are reprinting from *The Actuary* (April/May, 2014). Kurt explains the "Diderot effect," and how it can result in a dangerous escalation of complexity in our actuarial models. Quoting from Kurt, "these complex models often lead to a false sense of security among senior managers."

Brian Holland teaches us about "Unsupervised Methods: an overview for actuaries." How can you learn to model a business solution based on key parameters when you have not figured out yet what the parameters are? Given perhaps thousands or even millions of data points with hundreds of dimensions, how do you reduce the dimensions and cluster the data into meaningful groups; and what does it mean to train a model when you do not have the answers ahead of time to facilitate training?

Richard Xu and DiHui Lai continue this lesson with "Data Clustering and its Application in Insurance," where they teach us a procedure for clustering and useful measures for proximity; and then they apply the theory to an application involving the risks associated with foreign travel. They conclude with an observation that can help us better understand client behavior and market segmentation.

Rounding out our focus on clustering and regression, Geof Hileman and Claire Bobst give us an interesting application for the health insurance area with their article, "A Nearest Neighbors Approach to Risk Adjustment." They sum the algorithm in three simple steps (calculate distances, determine the neighbors, and weight results to determine new data points) and they tackle the non-trivial issues of determining distance for points more complex than (x,y), and how to determine an optimal value of K—the number of neighbors.

Jeff Heaton introduces another new modeling language to us. In "Agent Based Modeling with RePast Py," he takes us through a simulation with this free modeling extension to Python. If you remember John Conway's *Game of Life,* from 1970, this is taking the automaton concept and putting it on steroids. Jeff uses 10,000 consumer agents and 10 insurer agents and he has them interact to show how consumer demand may impact, and be impacted by, changing insurer product offerings.

As usual, Jeff provides his sample code on *www.GitHub.com/JeffHeaton/soa* for download. He also contributed a second article. This one is "Modeling with Python and Scikit-Learn." Jeff shows how you can use yet another free Python extension to run linear regressions, build and draw decision trees, and even model with ensemble gradient boosting. Ensembles are popular with Kaggle competitions, where world-class data scientists compete in predictive modeling contests.

At our home, our son likes to say the alliterative phrase "Prior Planning Prevents Poor Performance." Doug Norris shows he is the new master of alliteration though in "Parables and Prophecies Prevent Proper Predictive Prowess (Human Biases in Forecasting)." Despite the whimsical title, Doug makes a serious point: that we often tend to bias our own predictions. As Doug aptly explains, "Actuaries also like to be right, because being right feels good." His warnings are worth heeding; and his mitigation strategies make sense.

Our professional inclination to be right sometimes means we reject new ideas that sound less precise. I hope to temper the obsession with precision by continuing my introduction to fuzzy logic: "Warm and Fuzzy ... and Real! – Part 2." Here, I have tried to explain why you should be considering fuzzy logic and why the concept seems new even though it isn't. Along the way, I try to explain hedging, fuzzification, and defuzzification. The terms are not familiar to most people even though we perform them every day—probably thousands of times each day.

Finally, we want to let you know about an extension of the time limit for our forecasting contest. Doug Norris and Leslie Smith have composed an announcement for our website. We have copied it here in the Forecasting & Futurism newsletter for you. The articles in this issue may trigger some ideas to help you win $500 Apple Store Credit!

This is an issue packed with a lot of new and sometimes non-intuitive concepts and techniques. I won't pretend that every article is a quick read. Then again, the goal is not a trivial one. We want to continue to be known as:

精算师

Dave Snell

Dave Snell, ASA, MAAA, is technology evangelist at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at *dave@ActuariesAndTechnology.com*

CHAIRSPERSON'S CORNER:

Five Years Is A Lifetime (Personal Forecasting)

By Doug Norris

My first boss at Milliman had a saying that I've always enjoyed: "In business, five years is a lifetime." The intent behind the saying was that in five years, you can totally remake your career or your business.

My actuarial career has largely followed suit. Five years ago, I spent a plurality of my time helping health insurers implement individual and small group underwriting procedures. Today, the Affordable Care Act has made these strategies largely prohibited in the United States. Five years from now, one of my current specialties (strategies and tactics related to the Affordable Care Act's risk mitigation programs—affectionately referred to as the "3Rs") will no longer be relevant, with transitional reinsurance and risk corridors phasing out, and the overall risk of each market better known. So, how will I be spending my time five years from now?

Many of our business metaphors suggest a certain linear nature to our career paths; for starters, consider the "corporate ladder" that we are expected to climb one rung at a time. This is a comforting association because it suggests that once we've completed step N in our career progress, we can tackle step N+1 (and repeat until infinity, or until gold watch, whichever comes first). It also suggests that if we just put our head down and work hard, great things are destined for our future. The history of our career suggests a trajectory that will continue into the future. The "five years" adage expresses something quite a bit more complicated (and also a bit scarier). Do you mean that we can remake ourselves and that we likely have to remake ourselves?

Five years ago, I had very little knowledge of the Forecasting & Futurism section, which had just been rebranded from the original Futurism section name. However, in October of 2011 I was invited to present (as a pinch hitter) on the topic of complexity science at the SOA's Annual Meeting in Chicago. I had never presented at an SOA function, and had not dabbled in complexity science since my graduate school days. I took the opportunity, and the presentation went very well, but more importantly I found out that there were other actuaries like me out there, who were interested in advancing leading-edge techniques to improve our actuarial predictions. I was invited to run for section council (and won), which has led to two years (and counting) of great fun. It's been great exposure to new ideas and different practice areas. Plus, it has really broadened my perspective as to what our profession is capable of accomplishing.

The moral of this story? Take advantage of the opportunities that come your way. Be flexible and open to new ideas. Volunteer. Meet new people. Learn something new (even if—or especially if—you're done with actuarial exams). Find new uses for old skills. Surround yourself with people smarter than you, and people who will tell you when you're wrong. Don't be afraid to fail. Expect to fail. Don't be afraid to take a leap of faith every now and again; remember the SOA's slogan: "Risk is Opportunity."

As the new Forecasting & Futurism section chair, I have had the benefit of great role models—both Clark Ramsey and Alberto Abalo have set a leadership example that will be difficult to match, and they have been instrumental in ensuring that we are doing the right things as a section. In addition to Alberto, we have two others whose terms on the section council ended in October: Mike Lindstrom and Ben Wolzenski. All have been valuable members on the council, and their contributions have been both stimulating and numerous (including articles in this very newsletter).

For as long as I've been involved with the section, we have been blessed to have the SOA's Meg Weber and Christy Cook in support roles. We could not have grown our section into what it has become today without their tireless efforts. Both have recently left the SOA for new opportunities, although I'm hopeful that they will remain as "Friends of the Council."

Aside from being one of the best ambassadors for genetic algorithms around, Dave Snell has been the heart and soul of this newsletter for every issue since its inception in 2009, and the results are something to be proud of. He's always looking for new and innovative topics, and if you have an idea for a future issue of the section newsletter, I guarantee that Dave would love to talk with you. We are always looking for new authors.

Where will you be in five years' time? More importantly, where do you want to be in five years (and what can you do today to help get you there)?

Doug Norris

Doug Norris, FSA, MAAA, PhD., is a consulting actuary at Milliman in Denver, Colo., and a recovering professor (which suggests a future article: "Ten years is a lifetime"). He can be reached at *doug.norris@milliman.com*

Predictably Irrational, by Dan Ariely
The Hidden Forces that Shape our Decisions

Reviewed by Ben Wolzenski

This is considered a seminal work in the development of Behavioral Science. I found *Predictably Irrational* to be both interesting and illuminating, even if I did not always agree with the author's extrapolation of his findings to society at large.

I have included a brief summary chapter by chapter and described several of the author's experiments, almost all of which involved college students. It is worth reading the book to see them all, as well as his many engaging anecdotes.

INTRODUCTION – HOW AN INJURY LED ME TO IRRATIONALITY AND TO THE RESEARCH DESCRIBED HERE

The author describes how, after third-degree burns over 70 percent of his body, his rehabilitation treatment and limited ability to interact with others led to his scientific study of the reasons for human behavior. He posits that humans do not always behave rationally, and that therefore economic theory based on such rational evaluation of alternatives is flawed. In fact, he posits—and says he will demonstrate by results of scientific experiments chapter by chapter—that the irrational behavior is predictable.

CHAPTER 1 – THE TRUTH ABOUT RELATIVITY

Experiments show that humans evaluate options in relative, not absolute terms; that propensity can be used to manipulate decisions. In general, given a choice of options, we tend to choose one that is clearly better than another comparable one (which may be a decoy) rather than one which is hard to compare to other choices.

One of several experimental results is often cited: when a magazine offers an online-only subscription for $X, a print-only subscription for $Y (about twice $X) and a combination of online and print also for $Y, the combination was chosen by a large majority. The print-only subscription is effectively a decoy to make the combination look very good relatively. However, when only the online and combination subscriptions are offered at $X and $Y respectively, a large majority chose the cheaper online subscription; there was no print-only decoy to make the combination look more attractive.

CHAPTER 2 – THE FALLACY OF SUPPLY AND DEMAND

Arbitrary coherence, or anchoring, is described. "In life, we are bombarded by prices. … But price tags by themselves are not necessarily anchors. They become anchors when we contemplate buying a product or service at that particular price." As an example, a group of students were asked to write down the last two digits of their Social Security number. They were then to write down whether they would buy each of several items for that amount in dollars (yes or no) and then write down how much they would be willing to pay for each item. Those with the last two digits of their S.S. final digits from 00-20 were willing to pay far less than those with final digits from 80-99, despite the arbitrary nature of the different anchors provided to each group.

From price anchoring, the author moves on to behavior herding or self-herding. What's that about? We get in line for a restaurant because others are in line (so it must be good) or we repeat a purchase because we found it rewarding the first time, without comparing it to other substitutes.

The conclusion is that arbitrary coherence is a powerful influence affecting prices we are willing to pay, and that a purely rational valuation based on supply and demand is not as important as classical economics would postulate.

CHAPTER 3 – THE COST OF ZERO COST

This chapter described experiments in which items of little value were offered for free and were chosen over other items of much greater value that were offered at a small cost. Classical economics would have predicted that the better value would have been chosen. When we consider a purchase, we evaluate the upside and downside, but when something is free we feel there is no risk of loss, and since humans have an intrinsic fear of loss, there is an emotional reaction to free items that is not there at any other price.

CHAPTER 4 – THE COST OF SOCIAL NORMS

The author distinguishes between the worlds of social norms (friendly help, not compensated) and market norms (you get what you pay for). Kept separate, all's fine; when the two

collide or get mixed up, it can produce undesirable results. Lawyers asked to help the needy for $30 per hour refused, but when asked to provide free services for the needy, they agreed. The offer was insufficient under market norms, but accepted under social norms.

Companies are advised that they cannot have it both ways with their customers. They cannot treat customers like family (social norms) one moment but then impersonally (market norms) the next. They are cautioned not to diminish elements of social norms with respect to their employees, such as flex time, cafeterias, and health benefits.

CHAPTER 5 – THE POWER OF A FREE COOKIE

In an experiment, pieces of candy were either sold for a penny or given away free. More students took the free candy, as expected, but the number they took was less (1.1 if free, 3.5 if purchased). From this, the author concludes "that when price is not a part of the exchange, we become less selfish maximizers and start caring more about the welfare of others." By extension, this helps explain why everyone is reluctant to take the last of shared food items in a group meal setting.

CHAPTER 6 – THE INFLUENCE OF AROUSAL

To measure the effect of emotional state on judgment, a state of sexual arousal was chosen for the next experiments. Participants (men in college) were asked to answer a series of questions about potential sexual activities under two different conditions—what they thought they would do if they were aroused (but were not) and what they thought when they actually were aroused. The study clearly showed a wide and consistent difference in the answers. For example, when actually aroused, the desire to participate in odd sexual activities was 72 percent higher than estimated when not aroused. In a footnote, the author states "we can also assume that other emotional states work in similar ways."

CHAPTER 7 – THE PROBLEM OF PROCRASTINATION AND SELF CONTROL

Americans save less, borrow more. Why can't we save like we used to? Why don't we follow through on good intentions—saving, dieting, exercising—instead of putting them off? When groups of students were allowed different degrees of flexibility in setting deadlines for turning in multiple papers, those who had well-spaced deadlines did best. Those who gave themselves the maximum time to turn in all papers did least well. Everyone has a tendency to procrastinate, but those who recognize it and use available tools to commit themselves are most likely to succeed.

CHAPTER 8 – THE HIGH PRICE OF OWNERSHIP

Once we own something, we generally place a higher value on it. Why? The author posits three "irrational quirks" in human nature, and as usual gives interesting supporting anecdotes.

- We fall in love with what we already have.
- We focus on what we may lose, rather than on what we may gain.
- We assume other people will see the transaction from the same perspective as we do.

There is also the "Ikea effect"—the pride of ownership is directly proportional to the effort required to make or acquire something. And ownership can apply to ideas or opinions, too.

CHAPTER 9 – KEEPING DOORS OPEN

Subtitled (Why Options Distract Us from Our Main Objective), in this chapter experiments demonstrated that it is hard for us to give up options, even when we risk losing a lot by refusing to give up those options. By the "irrational impulse to chase worthless options," the author says, "we

CONTINUED ON **PAGE 10**

PREDICTABLY IRRATIONAL … | FROM PAGE 7

fail to realize that some [valuable] things really are disappearing." This is one of the "consequences of not deciding."

CHAPTER 10 – THE EFFECT OF EXPECTATIONS

Experiments involving beer and coffee showed that expectations definitely influence choices and perception. "When we believe beforehand that something will be good … it generally will be [perceived to be] good, and when we think it will be bad, it will be [perceived to be] bad." Further experiments (with Coke and Pepsi) used an MRI to show that brain chemistry was actually different if they knew the brand in advance.

"Expectations also shape stereotypes." And stereotypes can affect behavior … including that of those who are not part of a stereotyped group. A group of undergraduate students who took part in a scrambled-sentence experiment loaded with words such as Florida, bingo, and ancient, walked down the hall as they left the building slower than those whose scrambled-sentence test did not have words suggestive of the elderly!

CHAPTER 11 – THE POWER OF PRICE

In two studies about surgical procedures, volunteers who had only simulated surgery (it looked like the surgery had been done, but it hadn't) showed the same improvement as those who had the full procedure. These and other medical or pseudo-medical treatments were examples of how expectations influence physical experience, and the placebo effect.

Furthermore, an experiment involved giving two groups of volunteers a placebo that was supposed to be a new painkiller. One group was told the painkiller cost $2.50 per dose; the second group was told it cost $0.10. You can guess the result: the $2.50 group almost all reported pain relief, whereas only about half the $0.10 group did. The "new painkiller" was just a vitamin C tablet.

What do we think about the use of placebos? What if they work? Does cost saving by the use of generics produce poorer outcomes because of the price effect? Are placebo experiments that withhold treatment ethical?

CHAPTER 12 – THE CYCLE OF DISTRUST

This chapter was subtitled, "(Why We Don't Believe What Marketers Tell Us)." What better to exemplify that than to set up a table with a "Free Money" sign (and it really was), and see how many people stopped to pick it up? The answer: 19 percent stopped for a $50 bill, only 1 percent for a $1 bill. The typical comment was, "there must be a catch." There wasn't. In society, we can see how false marketing and other forms of dishonesty cause the erosion of trust over time. Another experiment showed that people's distrust is so deep that even statements such as "the sun is yellow" were mistrusted if attributed to corporations or political parties.

CHAPTER 13 – THE CONTEXT OF OUR CHARACTER, PART I

Through experiments and citations, the author concludes that there's a little bit of dishonesty in almost all of us (but not too much—stopped by our superego). Students who had a chance to self-report better test results than they actually earned did inflate their results a bit, but not when they were first asked to write down as many of the Ten Commandments as they could remember. That salutary effect was the same when the students were asked to sign a statement that the study fell under the university honor code.

CHAPTER 14 – THE CONTEXT OF OUR CHARACTER, PART II

Anecdotal evidence and one simple experiment showed that people find it easier to cheat or steal when their activity is one step removed from cash (e.g., take someone's soda but not their cash to buy a soda). After other examples, the author concludes: "None of this makes logical sense, but when the medium of exchange is nonmonetary, our ability to rationalize [dishonesty] increases by leaps and bounds!"

Ben Wolzenski

Ben Wolzenski, FSA, MAAA, is managing member at Actuarial Innovations, LLC in St. Louis, Mo. He can be reached at *bwolzenski@rgare.com.*

CHAPTER 15 – BEER AND FREE LUNCHES

In this final chapter, the author presents one more experiment that exemplifies the predictable irrationality of humans. He then concludes by contrasting personal and public policy decision-making based on traditional (rational) economics, with that based on behavioral economics, which offers the opportunity for "free lunches" that traditional economics says do not exist.

In an experiment, individuals' choice of (free) beer was influenced by how others would perceive them as well as by how much they thought they would like the beer chosen. The author considers this further evidence of irrational behavior, i.e., not maximizing personal enjoyment. (An alternative view is that an individual might rationally weigh his or her perception by others as more important than how much he or she would enjoy the beer.)

ANECDOTAL EVIDENCE AND ONE SIMPLE EXPERIMENT SHOWED THAT PEOPLE FIND IT EASIER TO CHEAT OR STEAL WHEN THEIR ACTIVITY IS ONE STEP REMOVED FROM CASH.

As an introduction to Behavioral Economics, *Predictably Irrational* is an excellent place to start.

Risk Management and the Power of Simplicity

This article first appeared in the April/May 2014 issue of the Actuary. It is reprinted here with permission.

By Kurt Wrobel

Actuarial science, like many other professions, has changed substantially with the introduction of sophisticated computer programs and greater access to more detailed data. This increased capability has contributed to the development of more sophisticated models that hold out the promise for more accurate models. In addition to increased accuracy, this computing power has the capability to increase efficiency by eliminating "manual" inputs and other processes that require human engagement. These advances have led senior managers to put more and more trust in the models and their predictive power.

The problem with this narrative—and the focus of this article—is the very real risk management costs associated with the increased complexity, efficiency and overconfidence in the predictive power of models. Although I still strongly believe in these tools, I also believe the effectiveness of these tools should be considered in relation to the costs of using them—particularly in light of new legislation that has made the underlying assumptions and historical data much less accurate in predicting the future. In addressing this question, I will discuss the costs of using more complex models and the blind spots that develop when managers put too much importance on models. I will conclude by offering approaches that offer simple solutions to accomplishing our chief task of managing risk in a complex environment.

Kurt Wrobel

THE DIDEROT EFFECT AND THE BUILDING OF MORE COMPLEX MODELS

In the 18th century, French philosopher Denis Diderot wrote "Regrets on Parting with My Old Dressing Gown." In the essay, Diderot discussed how he had to constantly upgrade his furniture and decor to match his new dressing gown. At the end of his essay, Diderot complained that his entire life and financial position had been made worse and more complicated because he had to match his surroundings with his dressing gown. The term the "Diderot effect" was coined to describe the dynamic of upgrading your material positions to match a single purchase.

Kurt Wrobel, FSA, MAAA, is chief editor of Health Watch, the newsletter of the SOA's Health Section. He can be reached at *kjwrobel@yahoo.com.*

The effect can be seen in the development of actuarial models. As one adjustment prompts another similarly elaborate adjustment, a model can quickly become complicated and substantially more difficult to follow and review. Considering the variability of the underlying assumptions that drive the model, the model could be far too complicated for the required task when a minor change in an assumption could produce a dramatically different result.

The chief problem is that this complexity is not considered in light of the enormous costs associated with the increased probability of error and the difficulty in creating an intuitive connection between an input and a result. Although one could argue the theoretical soundness of the model, the potential for a large systematic error that is not discovered increases greatly and is often not considered in the development of the model.

THE DRIVE FOR EFFICIENCY

With insurance organizations driving for increased efficiency, it is only natural that similar efficiency questions would be asked of actuarial and underwriting organizations. In general, the goal is to decrease manual work and replace repetitive human interaction with a model that can be easily automated. The classic example would be a comparison between an Excel-based rating model that allows the review of specific formulas and a "black box" model that eliminates human input and does not allow that same degree of review.

Although it is difficult to argue with the premise, this desire for increased efficiency often does not consider the systematic errors that can occur when a more manual approach is used. Instead of allowing the inherent checking or review that occurs in a manual process, a single error can be magnified and propagated through an entire block of business. Similar to the development of more complex models, this additional cost is often not considered.

OVERRELIANCE ON MODELS

People love the allure of models that promise to predict the future. By avoiding the inherent uncertainty and anxiety created by unknown future results, these models offer the promise of a more secure future. This narrative has been bolstered by books Competing on Analytics by Thomas Davenport and Jeanne Harris is one) that romanticize people using sophisticated data systems to improve business decisions and better predict outcomes.

The problem is that this simple narrative often leads to overconfidence when managers put too much trust in these models—particularly when attempting to predict a complex system. Although several case studies could be used, the experiences with the hedge fund Long Term Capital Management and the inappropriate use of the Value at Risk metric during the 2008 financial crisis offer overwhelming examples of the hubris of putting too much confidence in financial models. Instead of soberly taking a holistic approach toward the accuracy of the financial models, the managers in these cases used the models as justification to unknowingly take more risk.

SOME SIMPLE APPROACHES

In many cases, the advances discussed here can be well worth the additional costs and should be used, but the added risks associated with complex models need to be considered. From a risk management perspective, these costs can contribute significantly to systematic error that may not be easily mitigated through an intuitive knowledge of the model. In addition, these complex models often lead to a false sense of security among senior managers.

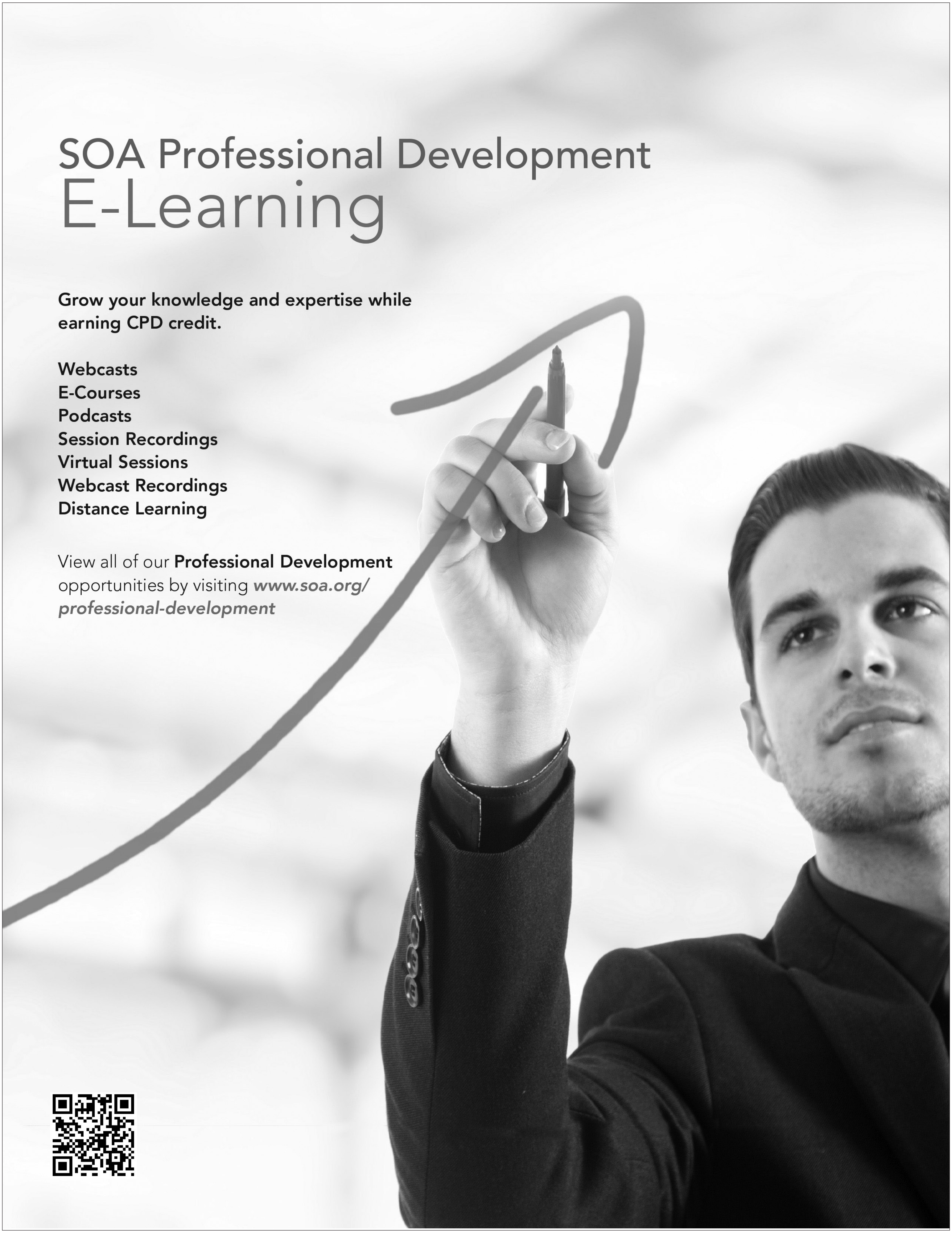
Before developing more complex models, I would suggest considering some simple rules to determine whether the additional complexity can be justified from a risk management perspective.

- If the underlying assumptions have the potential for substantial variability, the added benefit of complexity is much less than a more mature system where the assumptions are more stable. In short, if a single assumption change can have a dramatic impact on the result of a model, create a simpler model and focus your discussion on the key assumption.

- Even if complex models are used, a simpler model can still be used as an additional check to the overall reasonableness and accuracy of the model.

- We also need to consider decisions in light of the long-term viability of organizations that provide financial protection to people in the most vulnerable time of their lives. Unlike market researchers who analyze data to help to improve a company's website sales, we need to consider not just the short-term probability of an event using sophisticated data analysis. We also need to consider the long-term financial health of the entire system.

Most importantly, I think that we need to exercise wisdom. While this may include the use of complex models, in some cases, this may also include using qualitative judgment and consideration of other factors that could impact a business. As actuaries, we should be offering something well beyond a technical opinion; we should be providing a holistic opinion that ensures the long-term viability of our own organization as well as the broader insurance system. ▼


SOA Professional Development
E-Learning
Grow your knowledge and expertise while earning CPD credit.
Webcasts
E-Courses
Podcasts
Session Recordings
Virtual Sessions
Webcast Recordings
Distance Learning
View all of our Professional Development opportunities by visiting www.soa.org/professional-development

Unsupervised Methods:
An Overview for Actuaries

By Brian Holland

Actuaries might have some familiarity with unsupervised methods. In this article I'd like to focus on these methods, their differences from supervised methods, and some examples from actuarial practice that we might not generally know.

First of all: what are supervised methods? Who is supervising? You are supervising. Supervised methods involve you, the modeler, saying which observation depends on the other observations. Examples we all know are regression and classification. In regression, you say which number is *y* and which is x. That decision is the supervision. In classification you do the same thing: you know target categories, and are fitting items into categories based on characteristics. Linear regression, generalized linear models, and generalized additive models all fit into this category.

So what could unsupervised methods be? How could you do anything without saying what you're trying to model? That was my first question at least. Unsupervised methods have no labels with known meaning. Their goal is to find structure in the data. Think of the task as describing the space.

An example is clustering. Whatever is an *x* or a *y* or a *z*, you might first want to know if there are clusters. Are there clumps of items here or there? That question might be good to ask. There are some canned routines to compute clusters already in the Python language library scikit-learn for machine learning. Scikit-learn is the subject of Jeff Heaton's article in this newsletter. The scikit-learn documentation at http://scikit-learn.org/stable/modules/clustering.html includes a helpful comparison of types of clusters, types of algorithms to detect clusters, and the results of those algorithms.

Applications of clustering are right at hand:

- Actuarial models: how detailed should they be? Model points could be clustered.
    - Freedman and Reynolds (2008): "Cluster analysis: a spatial approach to actuarial modeling."
    - How much granularity do you need for premiums, assumptions, and models?
- Recommender systems, collaborative filtering
    - Customers who liked certain things might have found products that you also might like. Those customers form a cluster.
    - Clustering of types of objects based on similar characteristics.

Underwriting categories – we're clustering by appropriate premium level.

- If categories are already set: I'd say this is a classification problem: supervised.
- If categories are being developed, I'd say this is a clustering problem: unsupervised.

Facial recognition: which faces are similar?

In clustering the number of dimensions or attributes matters, whether you know what the attributes represent or not. For example, with only one attribute here, we see a couple of groups.

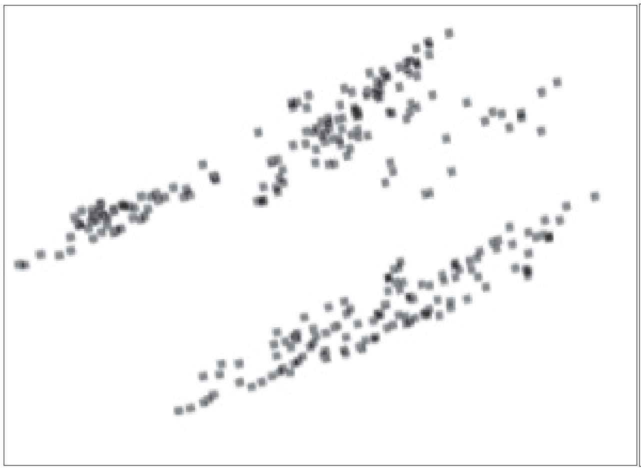
With two attributes the picture is much richer, and some groups in the 1-D example would get split into more, just from looking at it. The following image represents tilting the 2-D view forward in 3-D, to give a sense that there is a bit more going on here. In the bottom island in 2-D there is maybe a ridge of points which are closer together.

Brian D. Holland, FSA, MAAA, is director and actuary, Experience Studies at AIG. He can be reached at brian.holland@aig.com.

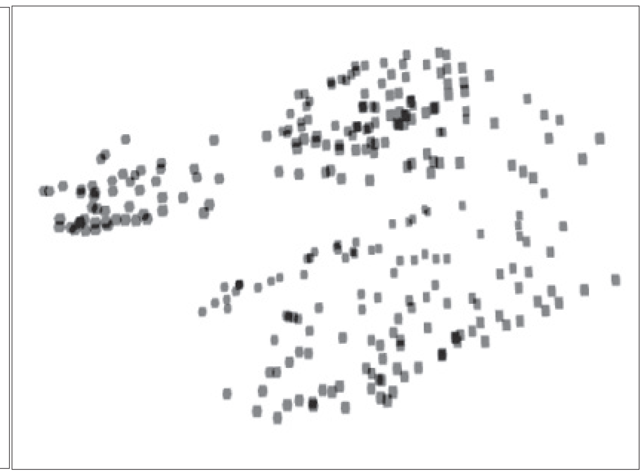
CONTINUED ON **PAGE 16**

UNSUPERVISED METHODS … | FROM PAGE 16

Points in 2-D



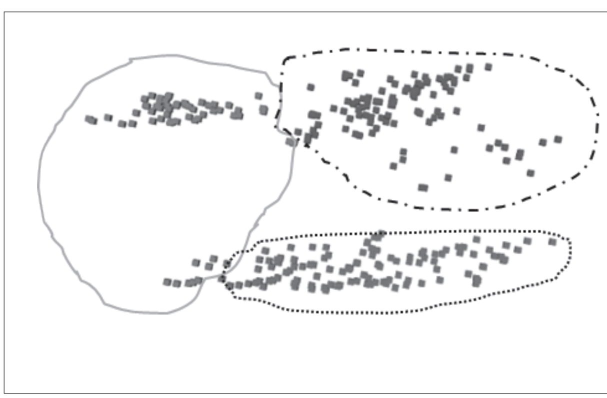
Points in 3-D, Tilted Away

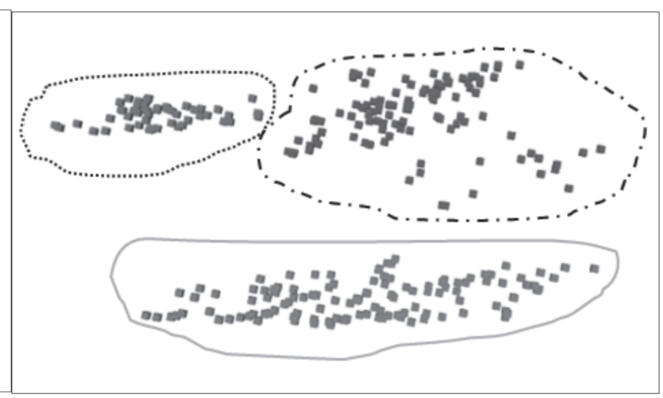


So how would you group these points into clusters? I recommend using well-studied and documented routines if in doubt. That way, the procedure is at least written up and known.

K-means: given number of groups (here: 3), pick k starting points to represent groups, assign each point to the nearest representative position (centroid), recompute the average after assignment, and do again. This is an iterative process. Here, the border between the three regions separates nearby points.

DBSCAN: emphasizes proximity of nearby points. The results here are generally the same classes, but there are some differences from K-Means. The lower cluster is all in one group.





The number of clusters we pick clearly depends on the number of dimensions we're considering. But what does "dimensions" mean anyway? An intuitive answer is that it means how many numbers are needed to describe the situation. For example, a line is 1-D even in 3-D space. As long as you can rotate and move your axes the right way, you can represent the whole line with one dimension. A financial example profit = income – expense. You might have three columns for the three numbers. If you know two, you know the third, so are there really three dimensions? There are not, not if you turn it the right way. The upshot is that you might not have to deal with as many numbers, columns, etc. as it appears at first, as long as each revised axis describes the right combination of original features.

DIMENSION REDUCTION

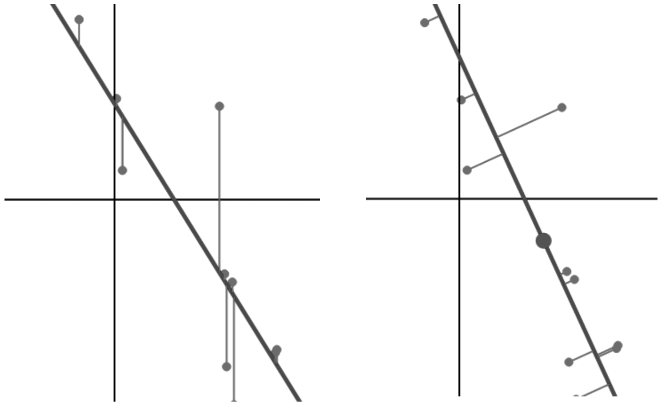
Why would we want to reduce the number of dimensions instead of using all the available information? I have two solid reasons for you:

1. Visualization: to see the main features. We can order the dimensions by variability along the new axes to see the main features.

2. Clustering: it is cheaper to calculate dimensions with fewer coordinates.

A technique called singular value decomposition (SVD) can be used to find and to order new dimensions. It determines the main dimensions or axes, which is those that pick up the most variance of the data, orders them, and quantifies the spread of the data around those dimensions. It differs from regression in that all coordinates (x and y for example) are treated the same. It minimizes squared distance to a line—a 1-dimensional subspace—not from a value y to a predicted value up or down the y axis. Then dropping that dimension from the new coordinates, there is one dimension left, and the procedure can be repeated until none are left. The sum of squared coordinates along a new axis indicates the variance along that axis, and orders the axes naturally.

Linear regression: minimize sum of squared distances from point up or down to the regression line.	SVD: minimize distance from point to the line. Here, the line goes through the average point, shown as a larger point.



CONTINUED ON **PAGE 18**

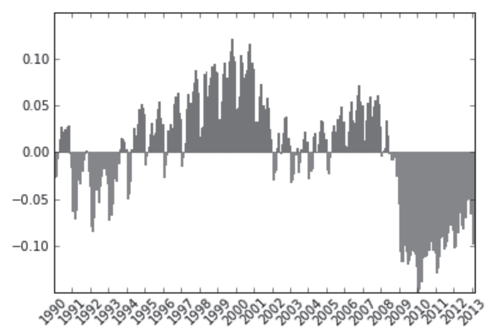
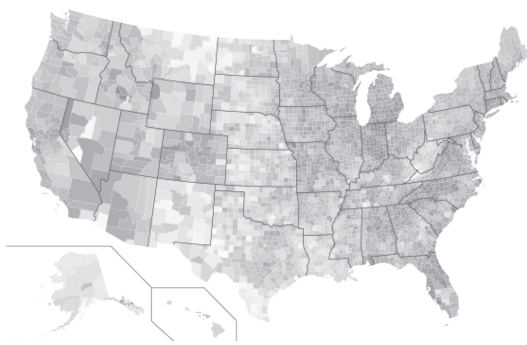
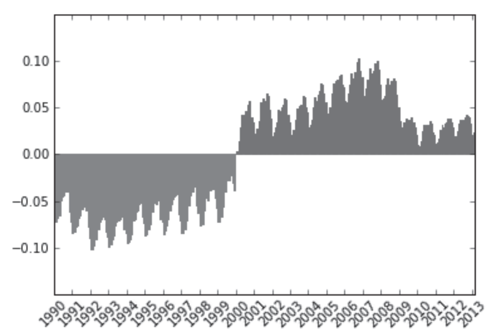
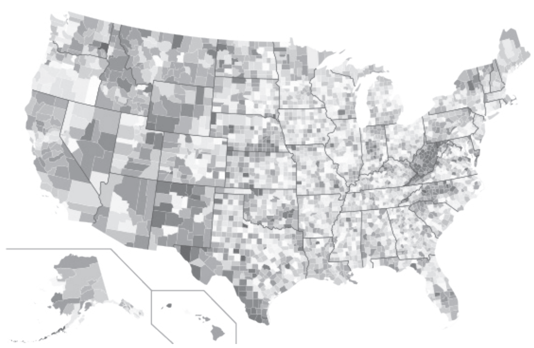
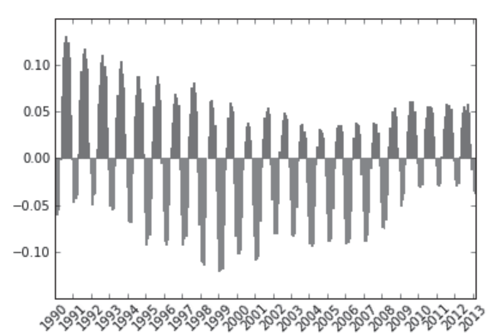
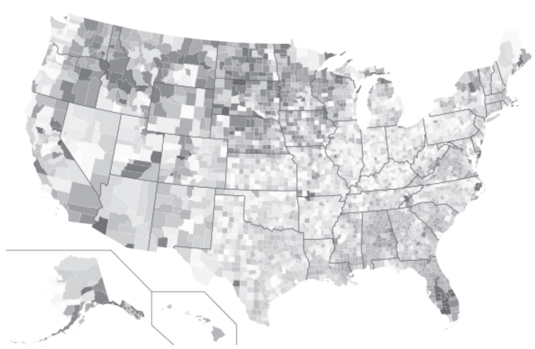
UNSUPERVISED METHODS … | FROM PAGE 17

When the decomposition is finished, the original data matrix *X* is represented as a product of three matrices:

$$X = U\ S\ V'$$

U and V are both orthonormal matrices. Their columns represent the new axes we could say. S is a diagonal matrix, with values decreasing down the diagonal. Multiplying these matrices amounts to taking the first column of U to scale each row of the result; the first column of V (transposed) to scale across the columns, times the first diagonal element of S for an overall scale level. Doing the same with the second columns of U and V and the second diagonal element of S.

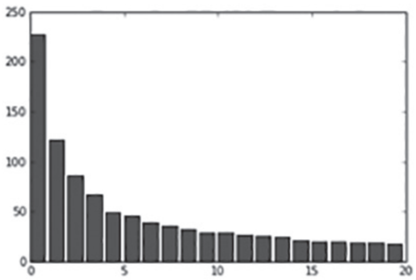
Can SVD help us find the main features in a larger dataset? To illustrate, I've tried out SVD on unemployment by county by month. I used unemployment because we all have some domain knowledge just from following the news. An animation of this detailed unemployment can be seen at *http://bdholland.blogspot.com/2013/05/visualizing-unemployment-by-county.html.* Some patterns are visible from watching the animation. SVD of the matrix of unemployment rates does show recognizable patterns. If we have X = U S V', a matrix with rows of months and columns of counties, then columns of U can be represented by time series; and columns of V can be represented as maps. We would make X by adding up layers: taking pairs of month and county vectors, blowing them out to make a matrix, and scaling them. The first three pairs are shown below. The first pair of columns of U, V is the familiar macroeconomic story: across most of the United States, there is worsening unemployment to 1992, improvement to the dot-com bust, then improvement to the mortgage crisis at which time there was a big spike in unemployment. The second layer is a regional correction. The third layer is my favorite: a mostly seasonal layer by date, with a map that clearly matches seasons. Note that the maps originally had red and green values: green for positive and red for negative. Values near white are near zero in any case.

Vector	Month singular vector	County singular vector
1		
2		
3		

CONTINUED ON **PAGE 20**

The scaling factors, the diagonal of S, are called singular values. They drop off by design, as later pairs are smaller or less important for describing the landscape. The singular values show the relative magnitude of the different layers, here for the first 20 of the nearly 280 layers. It is clear that much of the variability in unemployment by time and county is captured in the first three pairs of singular vectors shown above:

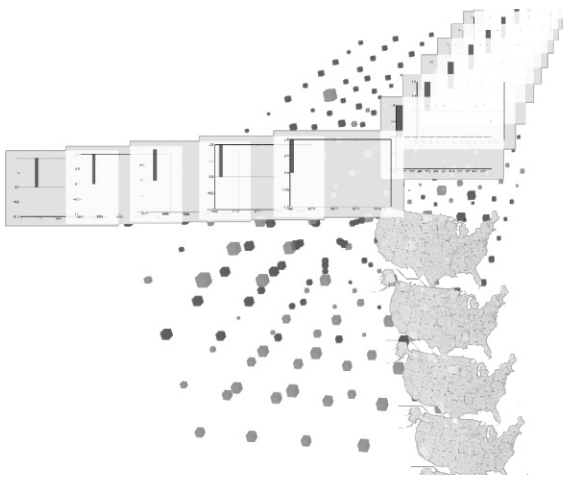
Singular values 1-20 of centered unemployment by county

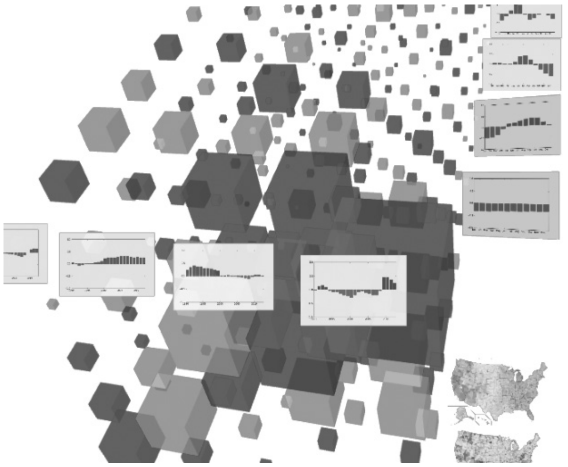


We can look at clustering of months as well by examining calendar months by the first three singular vectors. The clustering for different dimensions shown earlier was actually the calendar months. Below the months are connected as a time series, showing the three axes' meanings: the main singular vectors by county, represented as maps. It is now clear that the seasonal axis is the reason that the values bounce in and out through the seasons. The blue cluster (upper-left-most) is the period after the mortgage crisis. The macro axis, going down to the right, picks up the better and worse periods generally. From this exercise it is also clear what a weakness of this method is. We have to name the axis: if we're lucky, they make some sense, but they can be hard to explain. This issue can come up with matrix decompositions.

An actuarial application is the Lee-Carter mortality improvement model. Lee and Carter published their model in 1992. There have been changes since and that was some years ago, but that is the point: SVD has been used some time ago in actuarial applications. The model was not initially stated in terms of SVD, but Lee and Carter noted that the solution could be found with SVD. Note the language in the original paper: "… there are no given regressors." That is not supervised regression, but an unsupervised model. The authors effectively decomposed a matrix of mortality rates by age and calendar year, taking the first singular vectors for each of age and calendar year. For the projection of mortality rates, autoregressive integrated moving average (ARIMA) was used on the calendar year singular vector.

Higher-order singular value decomposition (HOSVD) is a logical next step beyond SVD. HOSVD means "higher-order" SVD. It goes by several names and has arisen in several contexts. It amounts to a way to decompose a tensor—effectively an array for actuaries—with more indices than the two indices of an array. The example above shows seasonality in each of the first three month-singular vectors. I decomposed a tensor of unemployment rates which was just a rearrangement of the same numbers into an array by calendar year, calendar month, and county. The unemployment rates by month (to the upper right), year (to the left) and county are shown on the left below for the top portion of the tensor. The decomposed tensor on the right shows the same county maps, but the calendar months show different patterns, and the calendar years are only the annual effects. The original tensor or array is replicated by scaling each combination of month, year and county singular vector triplet by the volume of the corresponding cube, and adding all such layers.





There are several closely related topics that are worth mentioning briefly:

SVD

X = U S V' possible for any matrix.

X: the (centered) data

U, V: their columns are called left and right singular vectors, respectively.

U, V are orthonormal, which also means we can see them as a rotation.

S: diagonal matrix, with values decreasing on the diagonal.

PCA: principal components analysis

V: columns are the *principal components*

U S: contains *principal component scores*

Covariance matrix of mean-centered matrix X is X' X / n = V S^2 V' /n since U^{-1} = U'

To my mind, the main point to remember about unsupervised learning methods is that they are used to find structure in data, without any domain knowledge of the source data or explicit modeling. They can be used to show the main features, which might be clusters of data, or high-level features. Clustering methods give mathematical support and convenience to functions that actuaries regularly perform.

PREDICTIVE MODELING SERIES

Data Clustering

And Its Application in Insurance

By Richard Xu and Dihui Lai

Predictive modeling is a very general term that includes many statistical algorithms to find relations in historical data for the purpose to predict future behavior. Some of these algorithms, such as data clustering, can be found useful for insurance applications.

Clustering analysis is a process of identifying patterns within a set of objects and grouping the objects with similarities into clusters. This unsupervised classification technique is ideal for exploratory analysis and is widely applied in fields such as object recognition and market segmentation. Additionally, this analytic method is also finding its way in supporting insurance, as more and more emphasis has been put on data driven decision-making. In this article, we are going to give a brief review on this method and demonstrate its power with an application.

INTRODUCTION

Data clustering groups objects into meaningful categories whose members exhibit similarities. Objects categorized in the same group are more similar to each other than to those in other groups. The similarity between objects is defined by a certain measure, such as distance between objects, dense areas of the data space, or other particular statistical distributions. Since each object is represented in a high-dimensional space, all these criteria have to be calculated in a multivariate method.

Clustering analysis can be broadly categorized as an unsupervised algorithm where data is not labeled. Explained mathematically, the input data have only variables $x_{i,j}$, but no target variable y_j (where i is the index for data fields and j index for data records). For situations where no knowledge about segmentation exists or it is impossible to label all data points for a large dataset, clustering analysis is a very powerful way to discover data structure and relationship.

As the name indicates, the main purpose of clustering is to help organize and describe the objects of interest. We can use the knowledge to naturally classify objects for deeper understanding, to explore the underlying data structure, or to organize the data. Clustering has been widely used in many fields to achieve these goals. Examples of clustering applications include identifying hierarchical systems in biology, information retrieval from the Internet, determining weather patterns in atmosphere and ocean for climatology, establishing book categories in libraries, and identifying common features and variations of disease conditions in psychology and medicine.

Business, including insurance industry, can also benefit from the application of clustering analysis. Very large amounts of information on current and potential customers have been collected. Clustering can be used to segment customers into a small number of groups for marketing activities. For example, market analysts can use cluster analysis to partition the general population of consumers into segments to better understand the relationships between different groups of consumers for marketing purposes.

CLUSTERING PROCEDURE

A simple clustering task can normally be completed in three steps: feature extraction, proximity measure definition and clustering/grouping.[1]

Feature extraction: This is a procedure of determining the features that best represent an object. For example, the most effective features to identify a person could be name, date of birth and gender. However, the effective feature could change depending on the question we are addressing. Considering the health condition of a person, the most useful feature might rather be his heart rate and blood pressure.

Proximity measure: Once objects are represented in their feature space, we need to determine the similarity measure between objects.[2] The most common measure is the Euclidean distance and is defined as

$$d_{Euclidean} = \sqrt{\sum_{i=1}^{n} (x_{A,i} - x_{B,i})^2}$$

for two objects A and B in an n-dimensional feature space

The more similar two objects are, the smaller the distance in their feature space. The distance measure can also be defined as the sum of the absolute differences between two objects along each dimension. This is known as Manhattan distance, represented by the formula. The measure is related to the walking distance (number of blocks) between two points in a city and is therefore also called city-block distance. Alternative metrics, such as Chebyshev, Mahalanobis or Canberra distance, could be useful measures depending on the nature of the problem.

Clustering/Grouping: Clustering is the major process where we determine how each object is assigned to certain groups. Hierarchical clustering and partition clustering are common methods.

Hierarchical clustering is an iterative process of connecting objects based on distance. For example, at the beginning, each object is considered to be a cluster of its own. Then the pair of clusters with the shortest distance (most similar objects) is linked to form a new cluster. This linkage procedure then continues on the newly formed clusters until no further cluster can be established. However, the resulting hierarchical structure does not provide a unique way of clustering and the decision on the number of clusters can be challenging.[3]

Comparing to hierarchical clustering, partition clustering does not produce hierarchical structure and is therefore less computationally intensive. The algorithm, however, requires the user to determine the number of clusters before the analysis which requires caution.[4] A well-known algorithm of partition clustering is the k-mean. The algorithm uses k pre-determined points in the feature space as the center of a cluster. Each object in the feature space is then assigned to the closest cluster center. The new cluster center is then calculated using the members in the current cluster. The procedure continues until convergence (when each cluster no longer has changing objects).

CONTINUED ON **PAGE 24**

DATA CLUSTERING … | FROM PAGE 23

APPLICATION: RISK SEGMENTATION ON FOREIGN TRAVEL

The risk involved in international traveling is of particular interest to life insurance companies. It is important to understand the possible risks associated with countries in a quantitative way. Here we investigated 205 countries, each of which is represented by 25 feature variables including life expectancy, HIV prevalence, Communicable Disease Death Rate, and GDP. (Table. 1).

Category	Feature Variables
Life Expectancy	Life Expectancy
Health	Maternal Mortality; Infant Mortality; Underweight Children; Adult Obesity; HIV Prevalence; Communicable Disease Death Rate; Physician Density; Sanitation; Drinking Water; Hospital Beds
Safety/Security	Traffic; Homicide; Military Conflicts; Foreign Deaths; Occupational Accidents
Environment	Carbon Dioxide; Particulate Matter Concentration
Infrastructure	Internet Users; Mobile Phone; Road Density
Economic	GDP Per Capita (PPP); Corruption; Education-Expected Years of School; Gini Index

Table 1: Features that are used as variables to describe the risk characters of a country. The data are collected from several different sources, including the World Bank, the U.S. State Department, the CIA World Fact-book, the World Health Organization, World Economic Forum and the United Nations. The missing values in the data set are imputed using the bootstrap expectation maximization (EM) algorithm.

To understand the possible risk ensembles formed between different countries, we explore the 25-dimensional feature space with iterative hierarchical clustering. We use Euclidean distance as a measure of the similarity between countries. A clustering of six groups is selected from the hierarchical structures constructed by the algorithm. With the aid of principal component analysis,[5] we mapped the feature space to a two-dimensional plane where we are able to see the relations between the countries (Figure 1). Countries from clusters one and four are quite distinct from the rest and are easy to separate by visual inspection. Please note that Figure 1 only plots clusters in two-dimensions. If we could include more dimensions, the clusters would appear more distinct.

The resulting clusters exhibit intuitive appeal. European countries such as Germany, Spain and the United Kingdom share a cluster with the United States, Australia and Hong Kong, whereas countries such as Afghanistan and Pakistan occupy a different cluster. The resulting analysis largely supports judgmental categorization of countries based on expected risk from foreign travel.

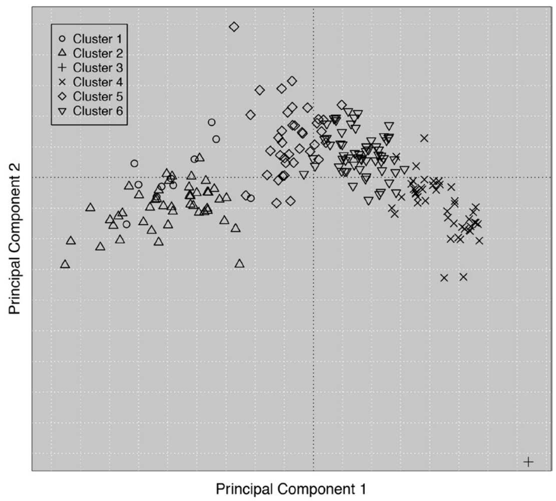


Figure 1: Projection of the feature space onto a two-dimensional principal component plane. Each dot represents a country and the color indicates the clusters that each country belongs to.

CONCLUSION

Data sets with high dimensions are normally difficult to understand. Data mining techniques can be helpful tools in exploring the underlying structures. This paper described how clustering methods could be used in an example within the life insurance industry. Further applications of the clustering method could help us better understand the client behavior, market segmentation, and customer classification.

REFERENCES

1. Jain AK, Murty MN, FLYNN PJ; Data Clustering: A Review; ACM Computing Surveys, Vol. 31, No. 3, 1999. https://ai.vub.ac.be/sites/default/files/data-clustering.pdf

2. Green PE and Rao, VR; A Note on Proximity Measures and Cluster Analysis; Journal of Marketing Research, Vol. 6, No. 3, 1969

3. Salvador S and Chan P; Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms; Tools with Artificial Intelligence, 2004.

4. DUBES, R. C. 1987. How many clusters are best?—an experiment. Pattern Recogn. 20, 6 (Nov. 1, 1987), 645–663.

5. Wold S, Esbensen K, Geladi P; Principal component analysis; Chemometrics and Intelligent Laboratory Systems, 2 (1987) 37-52.

Richard Xu

Richard Xu, FSA, Ph.D., is senior data scientist at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at *rxu@rgare.com.*

Dihui Lai

Dihui Lai, Ph.D., is data scientist analyst at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at *dlai@rgare.com.*

A Nearest Neighbors Approach To Risk Adjustment

By Geof Hileman and Claire Bobst

In the post-Affordable Care Act environment, health carriers are very limited in how much premiums can vary based on the risk levels of people seeking their insurance. In the absence of any other policy changes, this limitation would have incentivized insurers to avoid high-risk enrollees in favor of enrollees with fewer health conditions. This is clearly not a goal of the ACA. To minimize this risk, a risk adjustment program has been implemented to shift funds among insurers based on the relative risk of the people actually enrolled in their plans.

Risk adjustment models assign weights to various demographic and health-related categories according to the relative influence each factor has on an individual's cost. The weights corresponding to a given person's characteristics are then added up to determine their risk score. This risk score is normalized, meaning that a score of 1.0 indicates average health/risk, a score greater than 1.0 indicates worse-than-average health/higher risk, and less than 1.0 indicates greater-than-average health/less risk. These individual scores are then pooled to determine the average risk for a given group of people enrolled in a plan.

Behind the scenes, most risk adjustment models' weights are determined by regressions run on vast amounts of historical claims data. The method is quite effective and well-established techniques exist for its implementation, though it is still far from a "perfect" solution. The relatively low predictive power of these models is well-documented and, while this is mostly due to the variable nature of health care expenditure data, an opportunity potentially exists for a more powerful approach. By the nature of regression, all individuals in the sample are considered at once, and the incremental contribution of each of their characteristics to the average cost determines the outputted weights. Outliers, both high and low cost, are brought toward the mean, so that cost predictions for high-cost people tend to be too low and those for low-cost people too high, losing essential variation in the data.

A 55-year-old woman with multiple chronic illnesses, such as diabetes and heart disease, is unlikely to have similar costs to a 20-year-old male with no diagnoses. In regression risk adjustment models, they will, in however slight a way, have an influence on each other's predicted costs. To avoid this, what if, instead of running a regression on an entire dataset, we looked only at those people that most closely resembled the individual whose cost we were trying to predict:the people most similar in age, sex, diagnoses, prescriptions? In this way, we would only be considering the subset of the data most relevant to the individual of interest, and therefore the subset most likely to provide an accurate cost prediction for the individual.

A well-established algorithm, called *k*-Nearest Neighbors, can be applied to do just this. It consists of three simple steps:

1. Calculate the "distance" from the new data point to be classified to all the data points in the test set (note: dependent variable values are known for all points in the test set).

2. Determine the *k* data points with the shortest distances from the point in question. These are the "neighbors."

3. Average the dependent variable values of these *k* neighbors, weighting closer data points more heavily than those further away. This average is the approximated value for your new data point—the risk score.

The *k*-Nearest Neighbors algorithm is widely used in a variety of industries due to its simplicity, intuitiveness, and applicability to a variety of problems. Among these applications are the classification of breast tissue samples as malignant or benign based on a data set of known samples, the use of past weather data to create a stochastic weather generator and predict future weather in an area, or even audio fingerprinting, e.g., determining the identity of a song by comparing a short sample to a huge database of known samples (*k*=1 in this case). Nearest Neighbors is potentially applicable in most any situation in which past experience can be used to classify a new object.

Though the algorithm may seem simple and easy to implement, it is deceptively complex. In our application, we would like to predict the cost of an individual based on a set of people with known costs. Before we can do this, two main issues must be addressed. First, how do we determine the distance between two people? Our points aren't of the Cartesian (x,y) variety; instead, they are more complicated, consisting of a set of many different variables. We need to know how each relative difference and similarity in these variables impacts the difference in cost between two people, and therefore the "distance" between them.

The second issue is determining the optimal number of neighbors, *k*. The ideal *k* will minimize the error between the cost calculated by the algorithm and actual cost. *k* can be thought of as a smoothing parameter: it has to be large enough to smooth noise in the data but small enough to give an accurate estimate. A *k* value too small will be affected by noise, but a *k* too large takes into account irrelevant data points (at its limit, *k* is equal to the number of individuals in the sample and thus each individual is assigned the average cost, with closer neighbors weighted more heavily). Essentially, our choice of *k* has a tremendous impact on the accuracy of the Nearest Neighbors approach.

These two issues make apparent the work necessary to create a full-blown implementation of this algorithm. As such, our work has been of the proof of concept variety—investigating the idea to see if it has potential as an alternative approach to risk adjustment. To do this, we have been using R, a free statistical programming language, convenient in that it is both easy to use and provides open access to a huge number of packages written by programmers around the world.

We began our work in R by writing a script that would attempt to determine an effective distance formula. The idea here was actually to use a regression model, but with a subtle yet important difference from risk adjustment regressions: the model would return weights indicating the relative importance of each *difference* between two people in determining their *difference* in cost (how "far apart" they were).

THE K-NEAREST NEIGHBORS ALGORITHM IS WIDELY USED IN A VARIETY OF INDUSTRIES DUE TO ITS SIMPLICITY, INTUITIVENESS, AND APPLICABILITY TO A VARIETY OF PROBLEMS.

Unfortunately, this requires comparisons between every pair of people in a data file, a number that grows quickly with the size of the data. We realized that large amounts of computational power would be necessary to implement this regression, and that the results we were getting were unusable simply because we couldn't take enough comparisons into consideration. To temporarily deal with this problem we have been using weights from an already-established risk adjustment model as the distance formula coefficients. This is definitely not a perfect solution, and could affect the credibility of our results, but it at the very least provides us with a functioning distance formula for a proof of concept demonstration.

We then wrote a script in R to implement the *k*-Nearest Neighbors algorithm, and have been running various tests to look at how the results compare to that of a regression risk adjustment model. We have been using a data file containing 5,000 people, due to the fact that it takes about a minute for the NN algorithm to compare a given person to all 5,000 people in the file. Though a relatively small number of people to use, we've limited the sample to this size for now to avoid an even longer running time.

The program selects a given number of random people to classify from the data file using a specified starting seed. It then runs the Nearest Neighbors algorithm for a specified number of neighbors *k,* returning both the Nearest Neighbors error (NN cost – actual cost) and the regression error (regression cost – actual cost) for each random person.

CONTINUED ON **PAGE 28**

The algorithm was run for 40 random people in our data file for various numbers of neighbors *k*. The mean absolute error using nearest neighbors for these 40 people was calculated for each *k,* as was the percent of cases where nearest neighbors produced less absolute error in predicting cost than the regression model. No "best" value for *k* emerged, though more tests should be done to reduce the effect of data variability. All that is clear is that the number of neighbors should be greater than one and less than 50, which makes sense given the earlier discussion of *k*.

Error for 40 Random People at Various *k* Values

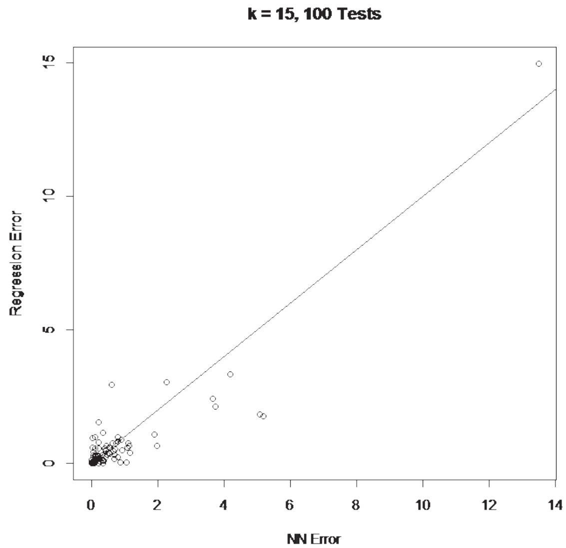
Total Number of Tests = 40		
k (number of neighbors)	NN mean absolute error	Percent of cases where NN abs. error is less than regression abs. error
1	0.9096	42.5%
5	0.7614	32.5%
10	0.7633	42.5%
15	0.7848	32.5%
20	0.7311	45.0%
50	0.8410	40.0%
Regression	0.6947	

Choosing an arbitrary value of *k*=15, 100 random people were generated and their cost predicted by the algorithm. The results indicated very similar absolute error for Nearest Neighbors and for the regression model, with NN producing less error for almost half of the people.

Error for 100 Random People, *k*=15

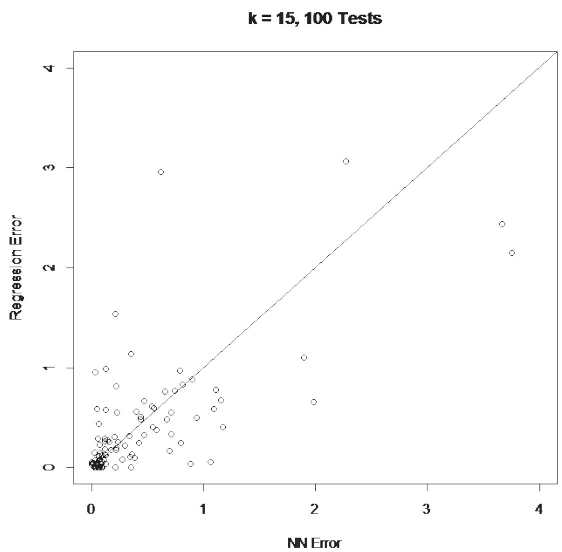
NN mean absolute error	0.70
Regression mean absolute error	0.61
Percent of cases where NN abs. error is less than regression abs. error	48%

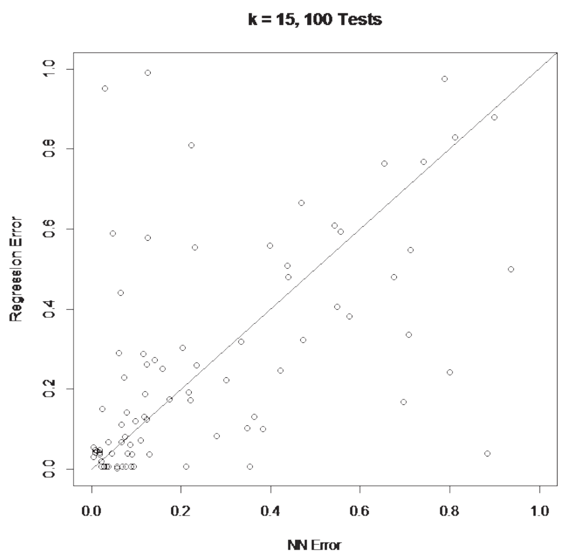
THE MAIN ISSUE WITH A NEAREST NEIGHBOR APPROACH … IS THE COMPUTATIONAL COMPLEXITY OF ITS IMPLEMENTATION.



The above plot displays Nearest Neighbors error versus regression error for each of these 100 randomly chosen individuals (one point indicates the results for one specific individual). One clear outlier is apparent, with large error produced by both methods. The vast majority of the data points, however, cluster around the unit square, providing a visual representation of the typical accuracy of the two models. The line y = x allows us to further compare the two: data points above the line indicate individuals for which the regression model produced more absolute error, while for points below the line Nearest Neighbors produced more error. This line seems to generally segment the data, supporting the fact that 48 percent, or about half, of the cases had less error produced by NN than by the regression model. The below plots zoom in on the data, the first on the interval [0,4], then on the unit square, to examine the majority of the data more closely and further illustrate this point.

Further experimentation has indicated that, as is the case with regression-based risk adjustment models, almost all values of *k* produce cost estimates that are too low for high-cost outliers. This again makes sense, but could be improved by increasing the size of the data file and thereby increasing the probability of finding neighbors that are





similar and have similarly high cost. Even with a larger data file, a smaller *k* will be ideal for these outliers, but this increases the extent to which their cost estimates are impacted by variability in the data. It is possible that the regression approach will remain preferable for these high-cost and/or rare-diagnosis people. On the flip side, Nearest Neighbors can produce very accurate cost estimates (0.1 error or less) for zero-cost people. This can be done using essentially any number of neighbors, due to the large number of people with no diagnoses, and thus zero or very low cost, in a given data file (specifically, zero-cost people make up 12.3 percent of our data file of 5,000 people).

We can conclude with several ideas for creating an effective implementation of the Nearest Neighbors algorithm. All generally revolve around the use of sufficient computing power. First and foremost, it is necessary to determine an optimal distance function, as having such a function that will allow definitive conclusions regarding how NN compares to traditional regression models. This will require processing a very large amount of data, as again the number of comparisons between people grows exponentially with the size of data. Not all people need be compared, but more comparisons will lead to a more accurate distance formula. Going along with this idea, the size of the test set should ideally be increased in hopes of improving predictions for high-cost outliers as well as for average-cost people.

The main issue with a Nearest Neighbors approach, both to risk adjustment and to the various other fields in which it is used, is the computational complexity of its implementation. We have to compare each new person to be classified to every other person in the test set, which simply takes a very long time. As such, there is a tradeoff between execution time and error: more data means slower execution time and less error, less data means faster execution time and more error. It is a fundamental issue. Some faster, modified Nearest Neighbor algorithms do exist, and it seems that these take one of two approaches: reducing the size of the data set in some way, or using some sort of tree structure to divide the test data into groups with similar characteristics. For our data set, we could take the first approach by reducing the number of variables involved, possibly by grouping together similar diagnoses (i.e., diagnoses in the same category but with different levels of severity). We did experiment with a simple tree structure, specifically by dividing our data into pre-defined demographic groups. Either approach could decrease the computational issues inherent in the problem.

CONTINUED ON **PAGE 30**

A NEAREST NEIGHBORS … | FROM PAGE 29

k-Nearest Neighbors is a potential alternative to the traditional regression approach to risk adjustment. Despite being a very simple algorithm, there are layers of complexity underlying its implementation, many potential questions to be raised and problems to be addressed. While a Nearest-Neighbors approach may not ultimately prove ideal for many risk adjustment applications, our preliminary evaluative efforts have suggested that this approach has potential for improving the predictive accuracy of risk adjustment algorithms.


Geof Hileman

Geof Hileman, FSA, MAAA, is director of actuarial studies with Kennell and Associates, Inc., in Raleigh, N.C. He can be reached at ghileman@kennellinc.com.


Claire Bobst

Claire Bobst interned with Kennell and Associates, Inc., in 2014, and is currently a senior mathematics major at the College of William and Mary.

Agent Based Modeling With RePast Py

By Jeff Heaton

Agent Based Modeling (ABM) can be used to simulate highly complex systems. ABM's are designed to model how participants in a complex system interact with each other. This puts ABM into the category of unsupervised learning. Unlike linear regression and generalized linear models (GLM) you do not fit past data and receive a prediction on future data. Rather, you set a model into motion and observe how different parts of the model interact over time. Additionally, ABM is always time series, as ABM always occurs over a defined time range.

ABM models attempt to deal with the chaos theory concept of the "butterfly effect" by modeling how a small change in one part of the model might have a huge change in another. In 1972, Philip Merilees stated the theory as, "Does the flap of a butterfly's wings in Brazil set off a tornado in Texas?" Consider if we simply wanted to model the profitability of an active and passive real estate investment strategy. The difference between the two approaches could be compared using historic data. Similarly, forecasts of the two models might use analyst predictions about market value and volatility. However, such predictions assume there is no interaction between the buyers and sellers. If many investors follow the more active investment strategy, does the increase in transactions affect real estate prices paid by practitioners of both investment strategies?

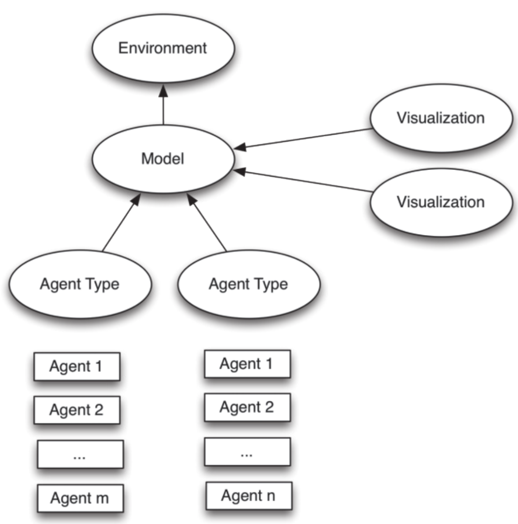
Agent based modeling attempts to consider these interactions. To construct an agent based model you create several different types of agent. To model real estate investment you would create agent types of investors following active and passive investment strategies. However, you would likely also create agent types for properties, builders, lending institutions, real estate agents/brokers, and many others. State variables are defined for each of these agent types. These variables would include cash on hand, property value, properties owned, and others. Additionally, the agents would form relationships, or links. An investor agent would form a link to every property that they owned. Finally, you would define actions. Actions are usually implemented in a programming language, such as Python, Java, C++, C# or R. Actions define how the agents affect each other. The action allows you to specify the effect that an agent purchasing a property has on all of the other agent types.

INTRODUCING REPAST

This article will introduce you to the Recursive Porous Agent Simulation Toolkit for Python (Repast Py), a free open-source ABM platform that is part of a suite of products that includes Repast Simphony, Repast for High Performance Computing and several language-specific instances of Repast. Repast Py allows the agent actions to be defined using a specialized scripting language based on Python 2.7. Repast was originally developed by David Sallach, Nick Collier, Tom Howe, Michael North and others at the University of Chicago. Repast is widely used. Google Scholar lists more than 50,000 citations and references. Additionally, Repast was used by Alan Mills for the "Simulating health behavior" SOA research project, sponsored by the SOA Health section.

Repast's architecture is composed of an Environment, Model, Agent Types and Visualizations. The Environment allows your agents to access supplemental information, such as files or a database. The model holds global model values and actions. This structure is shown in Figure 1.

Figure 1: Repast Py Architecture

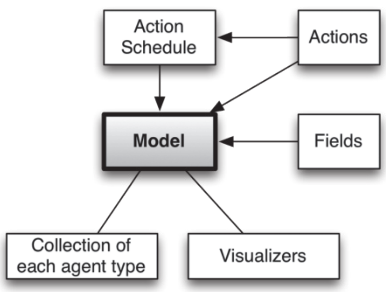


CONTINUED ON **PAGE 32**

AGENT BASED MODELING … | FROM PAGE 31

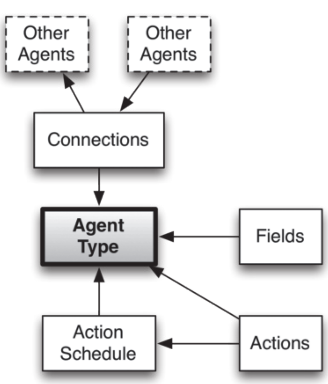
The model can hold actions and properties that are global to the agents. For example, you might store the prime interest rate at the model level. Agent types can also hold properties and these properties will be unique to each agent instance that is created. This structure is shown in Figure 2.

Figure 2: Models & Environments



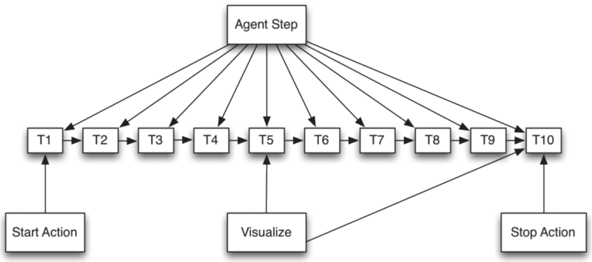
Agents are the real workhorse of an ABM model. One or more agent types are defined for a model. The model can have many instances of an agent type. Larger numbers of agents allow closer approximations to reality. The example provided for this article makes use of 10,000 agents. Agents contain actions that define how the agents interact. Agents can be competitive, cooperative or oblivious to other agents, depending on how their action code is constructed. Agents also form links to each other and maintain private and public properties. Figure 3 shows an agent type.

Figure 3: Agents



Agent based modeling is a simulation that runs in time. ABM time is expressed in a series of uniform intervals called ticks. Schedules are created that define the intervals that agent and model actions occur. Additionally, actions can be scheduled to only occur at the beginning or end of the simulation run. Visualizers are tied to intervals to define the granularity of their display. Figure 4 shows how actions are scheduled with the passage of time.

Figure 4: Passage of Time



CREATING A SIMPLE REPAST MODEL

For this article I created a simple ABM with Repast Py. You can download the source code to this article from GitHub, at the following URL:

https://github.com/jeffheaton/soa

The example is stored under the folder annual-2014 because this is an example that I presented at the SOA 2014 annual meeting. This example simulation can be used as a starting point for other simulations of your own.

This example seeks to model the following very simple scenario.

- Simple model of insurer response times to meet varying consumer demand for five insurance products.
- Two agent types: consumer and insurer.
- Consumers demand one of five products. Once demand is satisfied, consumer will cycle to the next product. (e.g., 1->2,…,4->5, 5->1)

- Insurers supply one product, and may retool when half of requests are unfilled.
- Initial set of insurers offer random products chosen uniformly.
- Initial set of consumers demand random products chosen uniformly.
- Model will track the rise and fall of the demand of each product on a linear plot.
- Initial setup will be 10,000 consumer agents and 10 insurer agents.
- Experiment with different insurer counts.

To implement this we create a consumer and insurer agent. The consumer agent has a property that defines what product the consumer is currently demanding. The insurer has properties that define both the product currently supplied, as well as the current cash on hand. Cash is not really used by the current model, however, it could be used as a performance visualization. The model is setup with each consumer agent demanding a different product. Likewise, each insurer agent is set to providing a random product.

Listing 1: Model Setup

```
# Cause the customers to demand random
products.
for consumer as ConsumerAgent in self.
consumers:
 product_num = Random.uniform.nextInt-
FromTo(0, 4)
 consumer.setProductDesired(product_num)
# Cause the insurers to offer random
products.
for insurer as InsurerAgent in self.
insurers:
 product_num = Random.uniform.nextInt-
FromTo(0, 4)
 insurer.setCurrentProduct(product_num)
```

Step actions occur at time intervals specified by the schedules. The step actions for the agents specify their interaction with the other agent. The step action for the consumer agent selects a random insurer and demands a product. If the insurer is unable to process this demand, then the failure is recorded, and the consumer demands no further products this tick. This is accomplished by the step action shown in Listing 2.

Listing 2: Consumer Step Action

```
# Choose a random insurer to obtain the
product from.
insurer_num = Random.uniform.nextIntFrom-
To(0, self.model.insurers.size()-1)
insurer = (InsurerAgent)self.model.insur-
ers.get(insurer_num)
# If the insurer has the product, then
obtain it.
if insurer.getCurrentProduct() == self.
productDesired:
 insurer.setCash(insurer.getCash()+1)
 self.requestFilled=self.productDesired
 self.productDesired = self.productDesired
+ 1
 insurer.setFilledRequests(insurer.get-
FilledRequests()+1)
 # Change our product desired, simply
cycle between 0 and 4.
 if self.productDesired>=5:
  self.productDesired=0
else:
# If the insurer does not have the prod-
uct, then record that.
 insurer.setFailedRequests(insurer.get-
FailedRequests()+1)
 self.requestFilled=-1
```

CONTINUED ON **PAGE 34**

The insurer step action evaluates failed requests from customers. If the insurer failed to fulfil 50 percent of the requests, then the insurer might retool and offer a different product. There is a cost associated with retooling. This process is shown in Listing 3.

Listing 3: Insurer Step Action

```
# Did we fail to fulfill any orders (pre-
vent div/0)?
if self.failedRequests>0:
 ratio = self.filledRequests / self.faile-
dRequests
 # Did we fail to fulfill 50% of the re-
quests?
 if ratio < 0.5:
  refit = Random.uniform.nextDouble()
  # Do we want to retool?
  if refit<self.model.probNewProduct:
   self.currentProduct =
      Random.uniform.nextIntFromTo(0, 4)
   self.cash = self.cash - 5
```

Python code is used to perform visualizations. Most visualizers in Repast are time-series line charts. The visualizer for this example shows the demand of all five of the products. Each line on the chart requires its own step action. Listing 4 shows the step action for Listing?? 1's (index 0) line.

Listing 4: Visualizer Step Action

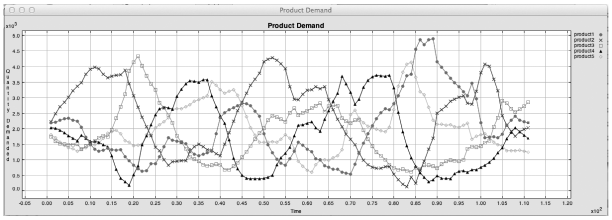
```
sum = 0
for consumer as ConsumerAgent
   in self.consumers:
 if consumer.getProductDesired()
   ==0:
  sum = sum + 1
return sum
```

The above code works by looping over all consumers and counting the number of consumers demanding product #0.

INTERPRETING THE RESULTS OF THE MODEL

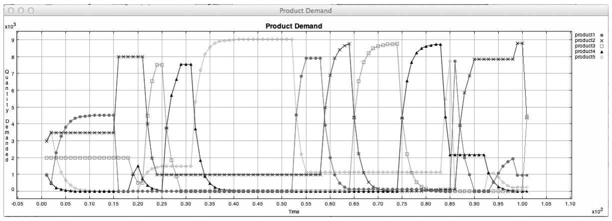
To see some of the insights that the simulation will provide, I created two visualizations. Each of these visualizations has 10,000 consumer agents. Figure 5 shows the simulation with 10 insureds.

Figure 5: Visualizer with 10 Insurers



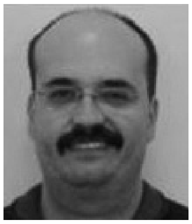
As you can see from the above, the demand for each product rises and falls somewhat smoothly. Because there are enough insurers, each product can be offered by one or two insurers at a time. If we force the model down to only two insurers we get much different results, as seen in Figure 6.

Figure 6: Visualizer with 2 Insurers



With fewer products being offered, the consumers are forced into narrow bands of demand. The transitions between products being offered are very sharp. Though the duration of demand for each product is somewhat random, the order in which the products are demanded in Figure 6 is mostly consistent.

ABM's are a great tool for forecasting the future through simulation. There are many considerations for building your own models. Often you will start the ABM somewhere in the past and tweak the parameters so that prediction matches reality up to the current date. The model then runs into the future providing predictions. Increasing the number of agents can provide more accurate results; however, it is important to ensure that the ratio of agent types makes sense. ABM's are a technology where you can start simple and increase the complexity of your model to handle increasingly complex situations. ABM can be an important part of your toolbox.



Jeff Heaton

Jeff Heaton, is data scientist at RGA Reinsurance Company and author of several books on artificial intelligence. He can be reached at *jheaton@rgare.com.*

Modeling With Python And Scikit-Learn

By Jeff Heaton

R and Python are the two most popular computer languages for data science, as reported by a 2013 KDNuggets survey. (KDNuggets, 2013) Both R and Python have a variety of data science frameworks available for them. These frameworks standardize the implementations of the many different models that data scientists use. This article will introduce the Scikit-learn (http://scikit-learn.org/) package for Python. (Pedregosa, et al., 2011) A similar package, called CARET (http://topepo.github.io/caret/index.html) is available for the R programming language. (Kuhn, 2008)

Scikit-learn is an open source machine-learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines (SVM), logistic regression, naive Bayes, random forests, gradient boosting machines (GBM) and k-means. Scikit-learn is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

The scikit-learn project started as scikits.learn, a Google Summer of Code project by David Cournapeau. The project's original codebase was later extensively rewritten by other developers. Scikit-learn is under active development and is sponsored by INRIA[1] and occasionally Google. Scikit-learn was used for a number of successful Kaggle (http://www.kaggle.com) competitions.

Kaggle is a platform for competitive data science that allows the top data scientists from around the world to compete on predictive accuracy. Kaggle has hosted a number of competitions of interest to the insurance industry. Allstate hosted a purchase prediction challenge, Liberty Mutual hosted a fire-loss challenge, and Practice Fusion hosted a challenge to predict diabetes in patients. The next article in this series will demonstrate scikit-learn modeling with data from a Kaggle competition.

INSTALLING SCIKIT-LEARN

There are many things to love about Python. However, multiple Python versions and installing new packages is not one of those things. Despite Python 3 being released in 2008, as of 2014 Python 2.x still has an active following. Backwards compatibility was severely broken when the switch to Python 3 occurred. Whenever using a Python example, it is very important to understand if you are using Python 3.x or Python 2.x code. The code in this article will work with either Python 3.x or Python 2.x.

Package management presents its own unique challenge in Python. Pure python packages can be installed with one of several package managers such as "pip" or "easy_install." Unfortunately, many of the Python packages contain compiled code based on C/C++ or even Fortran. This is the case with scikit-learn and some of the numerical packages it depends on. Fully describing how to install scikit-learn in Python 3 is beyond the scope of this article. I wrote a setup guide for Python that can be found at the following URL.

http://goo.gl/l94xQG

SCIKIT-LEARN BASIC LINEAR REGRESSION

Scikit-learn makes it very easy to switch between different model types. To start, consider a model to convert between Celsius and Fahrenheit temperatures.

```
x = [
 [-40],
 [10],
 [25],
 [30]
 ]
y = [-40,50,77,86]
```

The x-values represent the Fahrenheit input to the model, and the y-values represent the Celsius expected output. This is univariate data; it is also possible to use multivariate input data, as seen here.

```
x = [
 [1,2],
 [3,4],
 [5,6],
 [7,8]]
```

The above input contains two observations, or features, per sample.

For this example, we will use the first, univariate data set. Once we've defined the input and expected output for each observation it is easy to create and fit a model. The following code will fit a linear regression model.

```
from sklearn import datasets, linear_
model
model = linear_model.LinearRegres-
sion()
model.fit(x, y)
```

Now that the model is setup, we can query it with the "predict" command. To find out the Fahrenheit temperature for Celsius 10, use the following command.

```
print(model.predict(10))
```

The model will respond with 50 degrees. We can also predict a value not in the data set.

```
print(model.predict(15))
```

The model should return with 59 degrees.

We can also display the coefficients, RSS and variance for this linear regression using the following commands.

```
import numpy as np
# The coefficients
print('Coefficients: \n', model.coef_)

# The mean square error
print ("Residual sum of squares: %.2f"
%
    np.mean((model.predict(x) - y) **
2))

# Explained variance score: 1 is per-
fect prediction
print ('R^2 score: %.2f' % model.
score(x, y))
```

This results in the following output.

```
('Coefficients: \n', array([[ 1.8]]))
Residual sum of squares: 0.00
R^2: 1.00
```

Of course, the temperature conversion fit perfectly, so the RSS is zero and the R squared is 1.0.

SCIKIT-LEARN WITH A DECISION TREE

What if we wanted to use exactly the same data, only use a CART decision tree? Scikit-learn makes this very easy. The following code fits the temperature data using a regression decision tree.

CONTINUED ON **PAGE 38**

MODELING WITH PYTHON … | FROM PAGE 38

```
from sklearn import tree

model = tree.DecisionTreeRegressor()
model.fit(x, y)
print(model.predict(15))
```

Notice that the code is nearly the same? We always use the "fit" command to fit the model. Likewise, we always use the "predict" command to perform a prediction.

VISUALIZING A DECISION TREE

Scikit-learn allows visualizations of some of the model types. Decision trees are a model type that is particularly easy to visualize. To see how to visualize a decision tree, consider the following highly contrived data.

These observations represent individuals that applied for a particular type of auto insurance. The x vector contains their ages in the first column, gender in the second, and so on. For gender, one means male, and zero female. Martial has a value of one for married, or zero for single. The DUI and accident columns contain counts of each infraction. Finally, the y-vector contains a one for insured and a zero for declined.

```
features = ['age','gender',"marital",
"dui", "accident"]
x = [
  [16,0,0,0,0],
  [21,1,0,0,1],
  [42,0,1,0,0],
  [16,1,0,2,2],
  [34,0,1,0,1],
  [55,1,1,1,0]
]


y = [1,0,1,0,1,1]
```

Fitting this to a decision tree is easy enough, with the following commands.

```
model = tree.DecisionTreeClassifier()
model.fit(x, y)
```

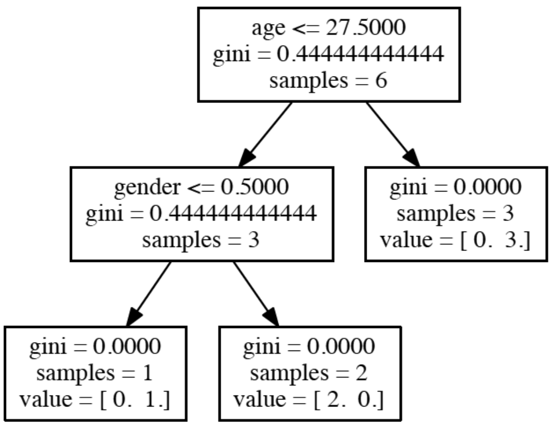
To visualize this as a tree, we use the following commands.

There are a number of "overhead" commands in the above code sequence. The main part to understand is that your tree will be written to "tree_pdf."

```
from sklearn.externals.six import
StringIO
import pydot
dot_data = StringIO()
tree.export_graphviz(model, out_
file=dot_data,feature_names=features)
graph = pydot.graph_from_dot_data(dot_
data.getvalue())
graph.write_pdf("tree.pdf")
```

This will result in the following tree.

The scikit-learn tree can be difficult to read, until you understand its format. Each node contains a single binary decision. If the condition is false, the tree will proceed to the left, similarly, the tree will proceed to the right if the condition is true.



The number of samples that support each tree node is displayed. As the tree descends, and specializes, the number of samples will decrease. Likewise, the Gini value should decrease as the tree specializes. Gini is specific to the CART algorithm and acts as a loss function to minimize.

Perhaps the most confusing line to understand is the "value." Only final decision nodes (leafs) will contain a value. Because we were classifying into two sets, there will always be two values in the "value" array.

The first number in the value array specifies the number of class 0, or decline, samples. The second number in the value array specifies the number of class 1, or accepts, samples. Ideally, only one of these has a value, and the other is zero. If this is the case, the Gini has the optimal value of zero. The output of the tree is usually interpreted to be the decision node's value with the most number of samples. For example, the right-most node on the above tree would output (be accept is this correct?), because there were three accept samples, and no decline samples.

SCIKIT-LEARN WITH OTHER MODEL TYPES

Scikit-learn supports many different model types. The following code would make use of a random forest.

```
from sklearn.ensemble import Random-
ForestClassifier
model = tree.DecisionTreeRegressor()
model.fit(x, y)
print(model.predict(15))
```

Similarly, the following code would make use of a Gradient Boosting Machine (GBM).

```
from sklearn import ensemble
model = ensemble.GradientBoostingRe-
gressor()
model.fit(x, y)
print(model.predict(15))
```

These advanced machine-learning models are overkill for this very simple linear data setup. For this completely linear, noiseless data set, the linear regression model is actually the most accurate.

OTHER FEATURES OF SCIKIT-LEARN

Scikit-learn includes many other features that assist in modeling. Model selection can be automated by trying many different model parameters. This slow process can be sped up using multiple processing cores on your computer. Scikit-learn also contains functions for feature selection, normalization, dimensionality reduction and many other common modeling tasks.

The next article in this series, titled "Titanic Pythonic Mortality Modeling" will look at the Kaggle dataset for the Titanic. This is a very simple mortality question, given statistics about the passengers, how accurately can we predict who survives and who perishes. The next article will demonstrate using Python and scikit-learn to accumulate, prepress and then model the Titanic data set.

CONTINUED ON PAGE 40

MODELING WITH PYTHON … | FROM PAGE 39

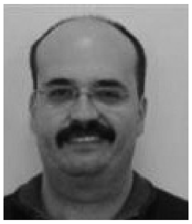
REFERENCES

KDNuggets. (2013, 8 1). *Languages used for analytics / data mining / data science* . Retrieved 7 31, 2014, from http://www.kdnuggets.com/polls/2013/languages-analytics-data-mining-data-science.html

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software , 28* (5).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011, 12). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* , 2825--2830.

Jeff Heaton, is EHR data scientist at RGA Reinsurance Company and author of several books on artificial intelligence. He can be reached at jheaton@rgare.com



Jeff Heaton

Jeff Heaton, is data scientist at RGA Reinsurance Company and author of several books on artificial intelligence. He can be reached at *jheaton@rgare.com.*

ENDNOTES

1 The **French Institute for Research in Computer Science and Automation** (French: Institut national de recherche en informatique et en automatique, **INRIA**) is a French national research institution focusing on computer science and applied mathematics.

Parables And Prophecies Prevent Proper Predictive Prowess (human biases in forecasting)

By Doug Norris

You work in the actuarial department of a smaller health insurance company, and recent claim experience is through the roof. The success of the firm hinges upon determining a reasonable forecast of where your members will seek services over the next 24 months, so that your negotiating team can properly weigh the costs and benefits of contracting with different provider groups. Fortunately, you've seen this very problem at your last company, and you know exactly where to look to solve it. Sure enough, the historical data show precisely what you expected. Following your recommendations, your company develops its utilization forecast, the contracting teams are fully armed, and everyone sets sail toward the brighter future ahead. All is well.

Two years from now, the company is suffering from even poorer claim experience. Fortunately, you've moved on to other opportunities, but your erstwhile colleagues are left to wonder: what went wrong?

There are many great things about being human, and advantages to not being built on an assembly line. However, our human experiences and preconceptions could be influencing our work as actuaries.

HOW DO OUR BIASES INFLUENCE OUR PREDICTIONS?

Randomness bothers us. Randomness suggests that there are things in the world that we cannot adequately explain. Randomness must be eliminated! As actuaries, we have a desire to explain. We have a desire to find patterns in the randomness (whether a true pattern exists or not). We also like a good story, preferably one with good guys and bad guys, one with a beginning, middle, and end, and one with a moral imperative. When one has a good "story," it is easier to find data that support those preconceived notions. It's also easier to filter out anything that doesn't support the story, even inadvertently.

Narrative bias occurs when we look for the convenient explanation—when we look for the good story. It's even better if we have our story prepared before starting our analyses, because having the story ready to go makes the analyses that much easier.

At the time of this writing, the risk mitigation components of the Patient Protection and Affordable Care Act (ACA) are of particular interest to commercial health insurers, and these provisions (risk adjustment, transitional reinsurance, and risk corridors) will ultimately have a significant impact on the bottom line for many carriers. The most difficult of these provisions to estimate is the risk adjustment program, because one needs to know both the calculated risk of the carrier of interest and the calculated risk of the overall marketplace.

Ultimately, the risk adjustment program is revenue-neutral across each marketplace, where issuers with healthier-than-average populations pay into the program to compensate those carriers with sicker-than-average populations. Here's the curious thing—nearly every carrier I've come across believes that its population is sicker than average, and are

CONTINUED ON **PAGE 42**

expecting to be recipients of risk adjuster transfer payments. Once you have the premise in your head, it's easy to find examples to support the story; Milliman's Health Cost Guidelines™ estimate that 19 percent[1] of a typical commercial population is sicker than the average member, so it's always easy to find anecdotes about "the hemophiliac" or "the transplant recipient" (and if you're not specifically looking for them, your chief medical officer will be happy to help)—there is plenty of support for the story. As for the portion of the population that incurs very few claims in the year (or none at all), they don't make any noise, so there's no narrative there.

Actuaries also like to be right, because being right feels good. This is a trait common to humans (of course), but applies to our profession in particular. We spend a good portion of our formative years passing a series of "right or wrong" credentialing tests. Actuaries are trained to believe that knowing the answer in advance is a very good thing.

Confirmation bias occurs when we seek out (overtly or otherwise) data and information that supports what we already believe to be true; it can also occur when we interpret neutral data in favor of our preconceived notions, or assign more weight to the data that support our opinions. As data become more and more plentiful, it becomes easier and easier to find the data that support what we already believe to be true.

Given that predicting the future involves a great degree of uncertainty, we typically rely upon our own experiences and history when setting our assumptions. Regardless of what data and evidence we have to support a given trend rate, we're always going to remember the last time that we were burned by an inaccurate trend forecast, and this will guide our opinion for years, possibly inappropriately. Typically, the more that is at stake (and the more that we have to lose), the more susceptible we are to confirmation bias.

WHAT CAN WE DO TO MITIGATE THESE BIASES?

The first step in remedying these human deficiencies is education and awareness. Now that you're aware of these biases, awareness should lead to education. One of my favorite recent books is *Thinking, Fast and Slow*, written by economist Daniel Kahneman, a Nobel Prize winner. The book explores a lot of what makes us irrational as humans, including the various heuristics and biases that we use every day (even when we're aware of these issues in advance). It's an easy read, and the exercises are certainly eye-opening (it's particularly humbling when Kahneman tells you in advance that you'll react irrationally, and then you follow suit exactly as predicted). In addition to understanding your own failings better, you'll better understand why irrational humans act in ways different than your rational models expect them to act.

When forecasting, it is important to take the extra time to actively seek out data and examples that dispute your hypothesis. As I said above, there's a lot of information out there these days, and it's not all going to support what you believe. Challenge your assumptions. Test the sensitivity of your forecast to each assumption. Seek out different approaches to modeling your problem. If diverse methods lead to the same outcome, it could be a nice sanity check on your results.

When we talk with stakeholders, we typically spend much of the time formulating a response. We want to impress those that we work with—to prove our value immediately by providing a solution. Always remember that "our solution" is based upon our past experiences and biases, and not necessarily on the current situation. Remember to listen to the stakeholders. *Really* listen to the stakeholders. You'll be surprised what you will hear if you listen, and you will end up with a more informed solution (and a more accurate forecast).

It's also important to quantify and communicate the uncertainty involved in our forecasts, with confidence intervals, significant digits, or other methods. Forecasting necessarily involves uncertainty, and to think otherwise (consciously or otherwise) is a recipe for disaster. Reminding yourself of uncertainty will also keep you cognizant of what could cause your projection to fail (and potentially identify biases underlying your work). Often, our audience demands certainty (or implied precision, which they may view as the same thing). Make sure that uncertainty is a part of your message; our job is to manage risk and to provide education on the impact of uncertainty (upside and downside).

Ultimately, no matter what cautions we take, we are only human. Peer review is a fundamental concept of actuarial work, and this needs to be more than just a compliance check box. Find someone who will really test or challenge your assumptions. If you can't find that person within your organization, seek an external audit. There have been a lot of unknowns in the ACA implementation, and some of the most interesting work I've done has come when insurers have called upon us to challenge their pricing assumptions. Beware of "groupthink" in these solutions, or cases where the dominant personalities control the outcome. One option may be a Delphi study, which our section has explored as a hedge against these deficiencies (a nice introduction to Delphi studies may be found in the Society of Actuaries' "Land this Plane: A Delphi Research Study of Long-Term Care Financing Solutions"[2]).

In the end, this is a human problem that will be with us long after we're retired. Instead of lamenting our collective deficiencies, take the opportunity to build and grow your skill set. There's nowhere to go but up! ▼

ENDNOTES

1. 2014 Milliman Health Cost Guidelines, Commercial Claim Probability Distributions, Table 1A (All Coverages).

2. Society of Actuaries (2014). Land This Plane: A Delphi Research Study of Long-Term Care Financing Solutions. Retrieved September 11, 2014, via *https://www.soa.org/Research/Research-Projects/Ltc/research-2014-ltp-ltc.aspx.*

Doug Norris

Doug Norris, FSA, MAAA, PhD, is a consulting actuary at Milliman in Denver, Colo. He can be reached at doug.*norris@milliman.com.*

Warm and Fuzzy … and Real! – Part 2

By Dave Snell

In our last issue, I introduced the idea of linguistic variables, which we all use in our daily lives, and which guide our decision process.

Most of us cross the street (or driveway or parking lot) each workday. When is it "safe" to cross the street?

> We might say it is "safe" when:
> There are no cars or other motorized or human-propelled vehicles as far as the eye can see; and
> the weather permits you to see far enough to ensure that anything that appears will not be able to overtake you before you reach the other side; and
> the path is not hazardous with potholes, slippery spots, very high winds; and
> there are no nearby intersections or driveways or buildings that might be hiding a vehicle close by but out of sight; or
> you could hear any such approach because the ambient noise is not too distracting; and
> the curb height is not dangerous;

... and dozens more contingencies I have not included here.

No wonder some of us come into work already tired! We have been multitasking so rapidly that it is a relief to finally be "safe" at our desks.

In this article, a continuation of my article last issue ("Warm and Fuzzy ... and Real!"), I will go into a little more detail on the reasons why you should be considering fuzzy logic, why the concept seems new even though it isn't, how fuzzy and "crisp" logic share many properties and how they differ in important ways on others. Along the way, I want to address a couple of key processes involved when using fuzzy logic: hedging, fuzzification, and defuzzification. The terms are not familiar to most people even though we perform them every day—probably thousands of times each day.

For those readers who consider intimidating formulas containing lots of Greek symbols no longer interesting, rest assured that you don't have to learn the formulas. Most folks drive fine without knowing how the car engine and transmission systems work. As Jeff Heaton showed in his companion article last issue ("Fuzzy Logic in R"), we have "vehicles" to make your application of fuzzy logic less tedious.

On the other hand, even though it is not necessary to know how the engine works to drive a car, virtually all race car drivers have an expert knowledge of how their engine, transmission, drivetrain, suspension, etc. work, and how they interact with each other. The more you know about the underlying theory and practice, the better you can be in your chosen field.

Learning some theory can also help us avoid "misuse" of a tool or technique. It used to be a common mistake for children who were hot on a summer day, to leave the refrigerator door open so that it would cool the room. Since a refrigerator takes heat from its interior and releases it through the fins at the bottom/back of the unit, this would have the effect of actually heating the room (and spoiling the food and raising the electric bill). Likewise, we should use the convenient tools built into packages for R and other programming languages, but understand when to use which tool.

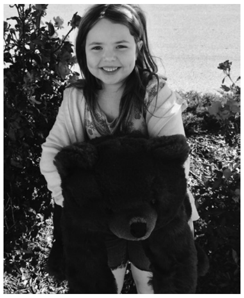
Let's start with some history. Our mathematical heritage is a strength; and we can be proud of the longevity of rules and principles developed hundreds, and even thousands, of years ago:

Fundamental Principle	Author and Creation Date
Geometry (Elements)	Euclid of Alexandria, c. 300 B.C
Laws of Motion, ($F = ma = m * dv/dt$)	Newton, 1642-1727 C.E.
Principia Mathematica (axioms, inference rules, symbolic logic – all mathematical truths can be proven)	Whitehead and Russell 1910-1927

These (and lots of other principles) have served us well for a long time. Over time, we came to accept them as absolute truths. Yet, some of these "truths" have turned out to be not quite true.

"FUNDAMENTAL PRINCIPLE" REVISITED	DISRUPTER AND DISRUPTION DATE
• A 0-dimensional point does not exist (or a 1-dimensional line; or a 2-dimensional plane) • and dimensions do not have to be integers	• Real world admission, (long overdue?), plus physical anomalies such as Möbius strips • Hausdorff–Besicovitch, 1918
Force = mass * acceleration ➔ F = m * dv/dt + v * dm/dt	Einstein, 1905 (as velocity gets very large, mass increases)
Gödel's incompleteness theorem shatters Principia Mathematica	Kurt Gödel, 1931

Our past knowledge can be an Impediment to understanding and modeling our world. Remember your first introduction to geometry? Your instructor said a line has length but no width. And the inquisitive child in you asked, "But I see it! How can it have no width if I can see it?" and then, in order to pass the course, you eventually conceded that it must be as the instructor said. Later, we learned that Euclid's geometry was a small subset of the vast number of useful ways to look at geometry. The Hausdorff dimension generalizes the notion of the dimension of a real vector space.[1] Einstein's "theory" of special relativity challenged and superseded Newton's second "Law" of motion. Gödel proved that for any set of axioms and inference rules proposed to encapsulate mathematics, there would in fact be some truths of mathematics which could not be deduced from them.[2]



Author's granddaughter, Amanda, with her warm and fuzzy friend.

Children look to fuzzy for comfort. They are naturally drawn to it … until we tell them to grow up and respect the sharp edges of life.

But life's edges are not so sharp! Our assumed "Laws" are continually being refined or replaced by better fits based on more careful observations of the real world. We utilize fuzzy logic every day. We just refuse (most of the time) to recognize it.

Lotfi Zadeh, a Professor at the University of California, Berkeley, is considered the founder of fuzzy mathematics, fuzzy set theory, and fuzzy logic. He published his seminal work, "Fuzzy sets," in 1965.[3]

Since the introduction of fuzzy sets, the classical Boolean sets have been renamed for convenience as "crisp" sets. Crisp logic is a new name for Boolean logic (George Boole, 1847). Crisp logic is binary in nature. Crisp set membership is always 0 (false, out) or 1 (true, in). Fuzzy logic allows interim values. Fuzzy set membership can vary from 0 (completely out) through an infinite interval of the real numbers (0.2, 0.67, 0.876, …) to 1 (all in). I described this in more detail in the article last issue, so here I'll just summarize some similarities and some differences between crisp and fuzzy logic:

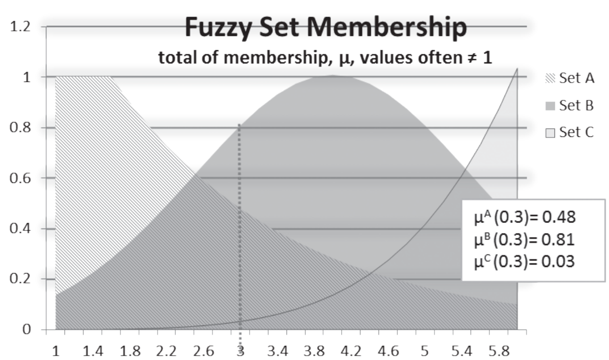
Relationships that hold for both Crisp and Fuzzy Logic	
Associative:	$(A \cap B) \cap C = A \cap (B \cap C)$, $(A \cup B) \cup C = A \cup (B \cup C)$
Commutative:	$A \cap B = B \cap A$, $A \cup B = B \cup A$
Distributive:	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
De Morgan's Law:	$\neg(A \cap B) = (\neg A) \cup (\neg B)$, $\neg(A \cup B) = (\neg A) \cap (\neg B)$

Fuzzy Rules
NOT x = (1 - truth(x))
x AND y = minimum(truth(x), truth(y))
x OR y = maximum(truth(x), truth(y))

CONTINUED ON **PAGE 46**

WARM AND FUZZY… | FROM PAGE 45

A key difference between crisp and fuzzy sets is that with fuzzy sets you can have varying degrees of membership in several different fuzzy sets at the same time; and the membership values do not have to sum to exactly one. You might say that X has membership 0.48 in set A, 0.81 in set B, and 0.03 in set C.



Of course, that oversimplification can lead to inconsistencies, so we also want to observe that sense prevails. If you have sets of underweight, normal, overweight, obese, and morbidly obese, then the same person could have 0.05 membership in normal, 0.80 membership in overweight, and 0.20 membership in obese (again, don't expect the sum to have to equal 1.0); but memberships of 0.5 in underweight, and 0.6 in morbidly obese, while 0.0 membership in normal seems to violate common sense.

Fuzzy Hedging

One way of addressing comparative memberships is through hedging. If a person has membership $\mu^{High} = 0.5$ in the set High, then without other data, we might assume the membership in Very High to be the square of that, since the square of a number between zero and one will be less than the original number:

$\mu^{VeryHigh} = (\mu^{High})^2 = (0.5)^2 = 0.25$ and $\mu^{AlmostHigh}$ to be the square root: $\mu^{AlmostHigh} = = 0.707$

Intuitively, this approach works well whenever you assume that the "very" set membership is more restrictive or exclusive than the "normal" membership, which in turn is more restrictive than the "almost" membership.

Fuzzy Logic Can Help You Get Quantitative Results!
Let's say we want to predict lapse rates as a function of interest rate and the unemployment rate. We might feel that if the unemployment rate is low and the interest rate is low, then policyholders will be inclined to keep their life insurance policies, and the lapse rate will be low. On the other hand, if interest rates are high, or very high, and so is the unemployment rate, then more policyholders will lapse their policies for cash needs or for arbitrage.

In order to predict lapse rates under these scenarios, we need to be able to determine the criteria that will allow us to say interest rates are high (versus low, normal, very high, etc.)

One approach is the old "crisp" approach—set precise ranges. Under that approach, we might say

3.001 percent to 5 percent is normal,
5.001 percent to 7 percent is high, and
7.001 percent to 9 percent is very high.

A disadvantage of this crisp approach is its focus on precision rather than accuracy. Do we really think there will be a big jump in lapse rates if the interest rate moves from 5 percent to 5.001 percent? A more realistic way to handle this classification situation might be to utilize fuzzy logic and to assume a Gaussian distribution, and this might mean that the move from 5 percent to 5.001 percent just increases the membership in High and decreases the membership in Normal.

Fuzzification

Fuzzification involves the classification of several quantitative values into linguistic variables. The programming language R[4] has several libraries available to handle fuzzy logic. One popular one is "sets." You can add this library to your R workspace with two simple statements:

```
Install.packages("sets")
```
#[5] this is a one-time download to your computer
```
Library(sets)
```
#this makes the many functions within "sets" available

Once you have sets available, define a universal set for computing the membership grades.

```
   sets_options('universe',
+  seq(from=1,  to=9,by=.5))
```
#note: the + signs, indicate a line continuation;

For this example, with just the three variables of Interest (int), Unemployment (unemp), and Lapse (lapse), we'll use membership values that fit into our defined universe. If the relative size of our variables were vastly different though, we'd normalize them for visual clarity in our plots.

Now that the housekeeping setup is done (at most three statements … not too bad so far!), let's define the three variables and the linguistic variables we are assigning to them:

```
vars<-set(
+     int=fuzzy_partition(varnames
+     =c(low=2,norm=4,hi=6,vhi=8),sd=1),
+     unemp=fuzzy_partition(varnames
+    =c(low=3,norm=4,hi=5,vhi=6),sd=.8),
+     lapse=fuzzy_partition(varnames
   +   =c(low=3,med=5,hi=9),sd=2))
```
#sd=standard deviation

Next, we'll create a couple of rules (in a real model, you would likely have lots more rules):

```
> rules<-set(
+     fuzzy_rule(int %is% low
+   && unemp %is% low, lapse %is% low),
+     fuzzy_rule((int %is% hi
+   || int %is% vhi)
+   && (unemp %is% hi
+   || unemp %is% vhi), lapse %is% hi))
```
#rules take the form fuzzy_rule(antecedent, consequent) i.e., condition(s) and the resulting implication(s) from the condition(s).

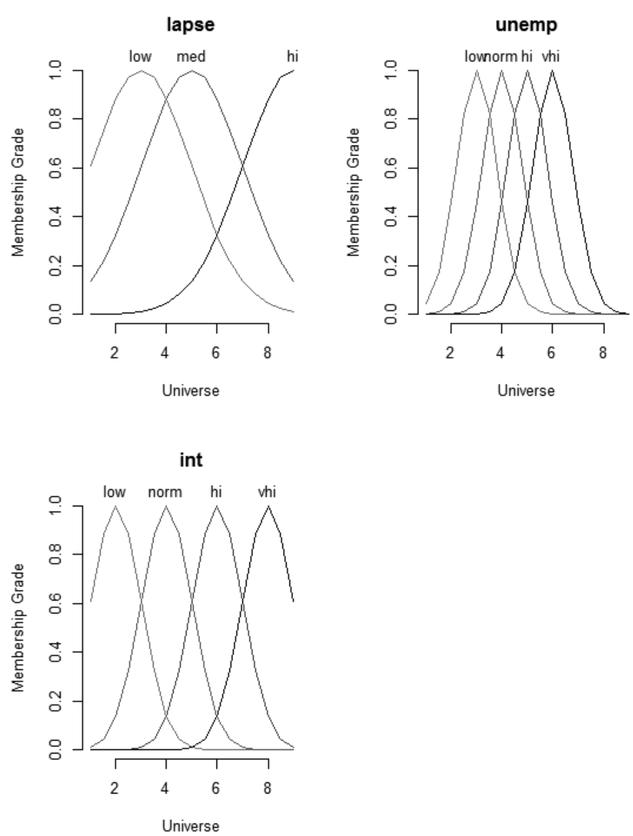
Finally, we connect our rules to our variables:

```
sys<-fuzzy_system(vars,rules)
```

All the hard work (for us) is now done. R can do the heavy lifting from here going forward. For example, say you wish to see a plot of your variables. Simply ask for it via

```
Plot(sys)
```

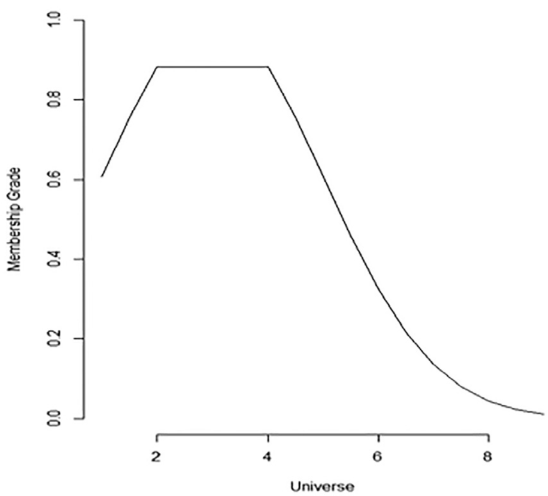
And voila! We get:



CONTINUED ON **PAGE 48**

It gets even better. Now, let's assume we want to see what kind of inference we get from an interest rate of 2.5 percent and an unemployment rate of 3 percent:

```
#fuzzify
fz_inf<-fuzzy_inference(sys,
+  list(int=2.5, unemp=3))
plot(fz_inf)
```



This shows us a graph of membership values for lapse rates at the various lapse rate possibilities. Clearly, the lapse rate is not likely to be high (9) since that membership, $\mu^{lapse}(9)$ is zero. The high membership values are clearly in the low area (low is centered at 3).

We have a graph, but we are not done yet. It still would be nice to quantify this.

Defuzzification

We'll defuzzify this graph to a single number with another function:

```
gset_defuzzify(fz_inf,'centroid')
[1] 3.496817
```

Our answer is a lapse rate of 3.496817; but how was that calculated? Let's make sure we don't just have the refrigerator door open!

The centroid method was used here, and the centroid can be computed by the integral

$$C = \frac{\int x g(x)\, dx}{\int g(x)\, dx}$$

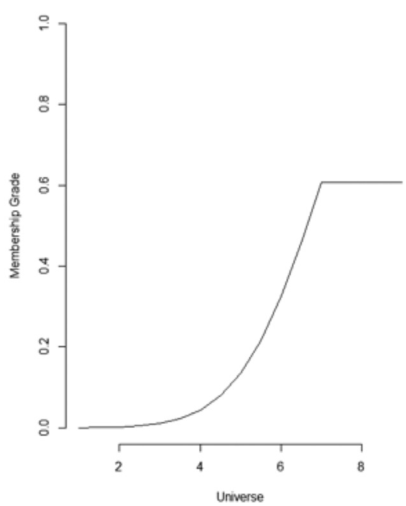
where the integrals are taken over the whole space, and g is the characteristic function of the subset, which is 1 inside X and 0 outside it. A physical interpretation of the centroid would be the point where you would place a pin and a string in order to balance the area. In this example, the centroid is probably appropriate, but this package offers a few more options for you for situations where the centroid might not be a good choice:

```
gset_defuzzify(fz_inf,'meanofmax')
    #returns 3 for this example
gset_defuzzify(fz_inf,'smallestofmax')
    #returns 2 for this example
gset_defuzzify(fz_inf,'largestofmax')
    #returns 4 for this example
```

Again, I urge you to investigate the many ways to defuzzify (there are several) to choose the best fit for your particular situation.

Before we leave this example, let's try one more input combination:

```
fz_inf<-fuzzy_inference(sys,list(int=7,
+   unemp=5))
plot(fz_inf)
```



This time we input a high interest rate and a high unemployment rate, and accordingly, the graph shows a high lapse rate is likely.

I'll quantify it according to all four defuzzification methods of this "sets" package:

```
> #defuzzify
> gset_defuzzify(fz_inf,'centroid')
```

#output is 7.322127

```
> gset_defuzzify(fz_inf,'meanofmax')
```

#output is 8

```
> gset_defuzzify(fz_inf,'smallestofmax')
```

#output is 7

```
> gset_defuzzify(fz_inf,'largestofmax')
```

#output is 9

Our range of returned values is 7 to 9, with a centroid of 7.322127 which is consistent with our rule that high unemployment and high interest rates would likely result in high lapse rates. If instead of just 3 to 9 percent, your product has lapse rates that might range from 5 percent to 75 percent, just normalize them to fit the universe we set for the other variables, then scale up after you have your answer.

The R vehicle is pretty easy to drive—especially for someone with a statistics background. Keep in mind though that the normal distribution is not always the correct function to use! If, instead of interest rates, we were fuzzifying wait times for a bus to arrive when we reach the bus stop at random times, the more appropriate probability distribution would be the exponential distribution.[6] Functions available include fuzzy_trapezoid (you specify the corners), fuzzy_cone (you specify the radius), etc. See Jeff Heaton's article (in Recommended Reading) for examples using other fuzzy shapes.

A Ferrari is not the best vehicle choice for hauling a trailer home; nor is a Humvee going to get you optimal fuel economy in city traffic. A Tesla, although quiet and powerful, would be risky for a trip over 300 miles through a remote area. Your knowledge of the subject matter of the problem and of the applicability of various statistical distributions will help you choose a better tool for the task at hand.

Fuzzy Logic can be a useful way to improve many actuarial models:

- It can be a closer match to the way humans think.
- Linguistic variables introduce both clarity and flexibility.
- Fuzzification can handle incomplete and inconsistent data.
- Rules sets can be cleaner and fewer in number.
- Defuzzification produces quantifiable results.

CONTINUED ON **PAGE 50**

WARM AND FUZZY… | FROM PAGE 49

Consider getting fuzzy to get more real!

Recommended Reading

- Shang, Kailan and Hossen, Zakir [2013] "Applying Fuzzy Logic to Risk Assessment and Decision-Making," CAS/CIA/SOA Joint Risk Management Section.
- L.A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," IEEE Trans. Syst., Man, Cybernetics, SMC-3 (1973), pp. 28–44.
- Snell, David,[2014] "Warm and Fuzzy … And Real!," F&F newsletter Issue 9.
- Heaton, Jeff, [2014] "Fuzzy Logic in R," F&F newsletter Issue 9.
- Klir, George and Yuan, Bo [1995], "Fuzzy Sets and Fuzzy Logic – Theory and Applications," Prentice Hall P T R, Upper Saddle River, New Jersey,1995.
- Ross, Timothy [2010] "Fuzzy Logic with Engineering Applications," Third Edition, John Wiley and Sons, Ltd., UK.
- Search for fuzzy logic on the SOA website for a current list of actuarial papers.

Dave Snell

Dave Snell, ASA, MAAA, is technology evangelist at RGA Reinsurance Company in Chesterfield, MO. He can be reached at *dave@ActuariesAndTechnology.com*

ENDNOTES

1 http://en.wikipedia.org/wiki/Hausdorff_dimension … and we can't even rest assured that we live in 3 dimensional space! String theory suggests perhaps 11 dimensions; or even 26-dimensional Bosonic space.

2 http://en.wikipedia.org/wiki/Kurt_G%C3%B6del

3 http://ac.els-cdn.com/S001999586590241X/1-s2.0-S001999586590241X-main.pdf?_tid=5f5a4f04-49eb-11e4-9b4c-00000aacb362&acdnat=1412223779_25c3ce3d1619da26ab8fa6440a8fcba9 "Fuzzy Sets," by Lotfi Zadeh, June, 1965, Information and Control, Volume 8, pp338-353.

4 In this example, I am using a free graphical user interface for R, called R Studio. Get it at www.rstudio.com.

5 Note that in R, # denotes a comment. Anything after the # on a line is ignored.

6 Doing Data Science, by Cathy O'Neil and Rachael Shutt], pp 128-129.

Forecasting & Futurism 4th Annual Contest

ACTUARIAL SCIENCE PREDICTIONS AND FORECASTING

One of our goals in the Forecasting & Futurism Section is to promote and educate actuaries with respect to innovating and cutting edge techniques for forecasting and predicting the future. In our newsletter, presentations, and webinars, we have explored a variety of topics, including predictive modeling, genetic algorithms and agent-based models, and Delphi techniques. As a part of these efforts, we are now inviting you to contribute.

Our fourth annual Forecasting & Futurism Contest is looking for you to explore and develop a technique that advances actuarial science predictions and forecasting in some fashion. This could be either:

- Investigating an interesting application of an established technique (such as predictive modeling or risk adjustment);
- Exploring an approach that has been advocated by the section (such as genetic algorithms, hidden Markov models, or Delphi techniques); or
- Advancing something entirely new altogether.

JUDGING CRITERIA

Entries will be judged on the basis of multiple criteria:

- (25 points max) How useful is the technique to the actuarial profession?
- (20 points max) How understandable is the approach (to an actuarial audience)?
- (20 points max) How easy would it be for another actuary to reproduce your work?
- (15 points max) How sophisticated is the technique (or extension) developed?
- (10 points max) How flexible is your technique? Could it easily apply to other applications?
- (10 points max) How creative is your approach?

Typically, we award an iPad to the winner of our annual Forecasting & Futurism contest. However, many of our audience already have an iPad, and there is a considerable delay between now and when the prize will be awarded (inviting the possibility that a new gadget will be released in the meantime). Therefore, the winner of this year's contest will receive $500 in credit to spend at the Apple store on any gadgetry of their choice (including perhaps an iPad).

ENTERING THE CONTEST

To participate in the contest, you must be a member of the Forecasting & Futurism Section at the time you submit your entry. If you are not yet a member of the section, you may enter the contest by first joining the Forecasting & Futurism Section.

The deadline for entering this year's contest is Jan. 31, 2015; please submit completed entries to Leslie Smith at lsmith@soa.org. We will confirm receipt of your entry via email. If you do not receive an email confirmation, please resend your entry.

If you have any questions, concerns, or compliments on the contest, please contact Doug Norris at doug.norris@milliman.com.

RULES

Entrants must be current members of the SOA Forecasting & Futurism Section.

The Forecasting & Futurism Section Council reserves the right to not award any prize in the event all entries are wide of the mark.

The Forecasting & Futurism Section and the Society of Actuaries may choose to use information about any of the entries submitted in publications or other venues of the SOA without further involvement of the entrant.

The Society of Actuaries reserves the right to substitute the cash equivalent value of the contest prize.

The contest winner is responsible for taxation issues as they are appropriate to his/her region. ▼

SOCIETY OF ACTUARIES Forecasting and Futurism Section

NONPROFIT
ORGANIZATION
U.S. POSTAGE
PAID
SAINT JOSEPH, MI
PERMIT NO. 263

Forecasting & Futurism

NEWSLETTER

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