Data and code to build district-month predictions of future violence in Afghanistan

Posted on **April 12, 2013**

Earlier this week, I posted an article about building predictions of future levels of violence at the district-month level for Afghanistan. Here is there data and the code is below. Note that it takes about 30 minutes (not 5) to run. Also note that it generates LOTS of errors since many of the time-series have long runs of 0's, but eventually these get predicted to be 0's, so it's all good. Let me know if you would like the province- or country-level data or code. Efficiency wasn't really my goal with this code (it's crude and clunky but achieves good predictive accuracy), the loop structure is slow but should be pretty easy to follow. Please let me know if anything doesn't make sense.

```
1   ##programmer: JEY
2   ##2/28/2013
3   ##District-month forecasts
4   ##note: this takes about 5 minutes to run.  It throws lots of
5
6   rm(list=ls())
7   library(foreign)
8   library(forecast)
9   library(sos)
10  data<-read.dta("raw_AFG.dta")
11
12  month=0
13  error.naive=0
14  error.arfima=0
15  start=580
16  end=627
17  for (k in start:end){
18    month<-k
19          error<-matrix(data=0, nrow=317, ncol=3)
20          colnames(error)<-c("true", "naive", "arfima")
21                  for (i in 1:317){
22                          test<-subset(data, newid==i &monthly<r
23                          a<-test$count
24                          unif<-runif(length(a), min=0, max=.1)
                            a<-a+unif #this minor addition allows
                            fit<-arfima(a)
```

```
                        y_hat<-forecast(fit, h=1)
                        y<-as.vector(y_hat$mean)
                        error[i,3]<-y
                        true<-subset(data, newid==i &monthly==
                error[i,1]<-true$count
                naive<-subset(data, newid==i &monthly==(month-
                error[i,2]<-naive$count
                }
        error<-round(error, digits = 0)
        error.naive[k-(start-1)]<-(sum(abs(error[,2]-error[,1]
        error.arfima[k-(start-1)]<-(sum(abs(error[,3]-error[,1
}

error.naive
error.arfima
compare<-cbind(error.naive, error.arfima)
dif<-as.matrix(error.naive-error.arfima)
results<-cbind(compare, dif)
colnames(results)<-c("error.naive", "error.arfima", "differenc
write.csv(results, "results_district.csv")
```

gistfile1.r hosted with ♥ by GitHub view raw

Posted in **Uncategorized**

The Effects of Intra-state Conflict on Interstate Conflict: An Analysis of GDELT

Posted on **April 12, 2013**

The release of the GDELT dataset has been receiving a lot of attention, and rightfully so (see Foreign Policy's write up here, Jay Ulfelder's write up here, and Phil Schrodt and Kalev Leetaru's official release paper here). Last week, I posted a chapter of my dissertation that used GDELT to build predictions of violence in Afghanistan (see the posts below). Below is a chapter that I wrote that uses GDELT to provide a rigorous, dyad-month level analysis of the effects of domestic conflict on interstate conflict.

The Effects of Domestic Conflict on Interstate Conflict: An Event Data Analysis of Monthly Level Onset and Intensity

Posted in **Uncategorized**

A Tale of Two Ph.D.s

Posted on **April 7, 2013**

On Friday, Slate published an article called "Thesis Hatement: Getting a Literature Ph.D. will turn you into an emotional trainwreck, not a professor", written by Rebecca Shuman. Since I don't know Shuman or anything about a humanities Ph.D. program, I'll withhold judgments that I would otherwise be inclined to make. I just figured I'd share a bit about my experience getting a Ph.D., since it is pretty much the exact opposite of Shuman's.

I began the Ph.D. program in the department of political science at Penn State in the fall of 2009 with no prior graduate schooling. Penn State, like most major universities, covers all tuition for students accepted into the Ph.D. program. It also provides a stipend that covers the basic cost of living. Early on in my first semester, I told the faculty that my intention was to finish the Ph.D. and then work for either the government or enter the private sector. Reactions ranged from neutral to highly supportive.

Last month, I completed my Ph.D., a little over 3 and a half years since I started. During that time, I took nearly a dozen methodological courses, four of which were outside the department but still fully funded. I taught an undergraduate class, did outside consulting work for the government, presented at conferences, and got a few publications. I also wasted a hell of a lot of time, and not in Shuman's "grad school is a waste of time" sense, but in the "going to bars to play pool and drink $5 pitchers of Bud Light" sense, meaning that I certainly could have been more productive.

The training I received as a Ph.D. student qualified me for a host of jobs, ranging from think tanks to government to academia to the private sector. In January, I accepted a job as a data analyst with Allstate Insurance and started in March.

I must acknowledge that I was lucky, since my department gave me (and the other students in the program) considerable freedom and support to pursue whatever interests. I also had an incredibly good advisor. I have no idea if my experience is indicative of other social science Ph.D. program and am not claiming that it is. All I know is that for me, the process was rewarding, fairly fast, and led directly to a good job.

Ok, so I lied – I will pass judgment on one point Shuman makes. She claims that a humanities BA is among the most hireable, which is just flat out wrong (note that the

supporting article was written in 1997, when Zuckerberg was in middle school, Jobs was just beginning his second stint at Apple, and Bieber was 3. A few things have changed since then.) If you don't believe me, how about a friendly wager? Call up the career services of a few major universities. Ask them if an undergraduate majoring in computer science with a minor in economics has a better chance of getting a job than an undergraduate majoring in philosophy with a minor in English literature. I'll bet any amount of $$$ that they pick the former.

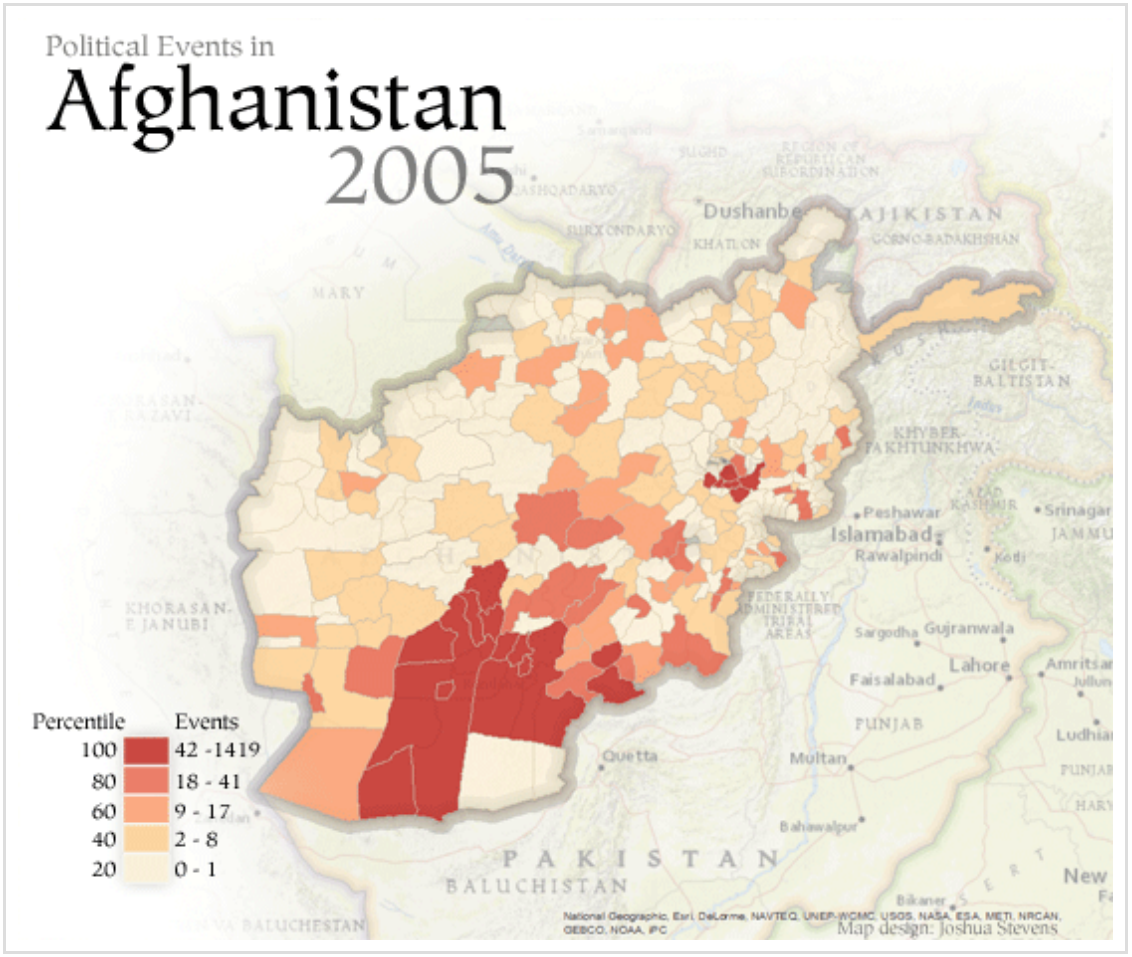
Posted in **Uncategorized**

Using GDELT to forecast violence in Afghanistan

Posted on **March 31, 2013**

The Global Dataset of Events, Location, and Tone (GDELT) is a new, 230 million (and growing daily) is the first ever machine-coded political event data dataset to provide information on event location. For those attending ISA, Kalev Leetaru and Phil Schrodt will be formally introducing the GDELT dataset. The full dataset will be publicly available soon, but for now you can access an older version here.

From a forecasting perspective, the benefits of a machine-coded dataset updated in (near) real-time that provides specific latitude-longitude coordinates are numerous. In the first ever empirical analysis using GDELT (pdf of paper –> "Predicting Future Levels of Violence in Afghanistan Districts with GDELT"), I build an empirical model that predicts the level of conflict at the district-month level in Afghanistan. Below is .gif that Joshua Stevens built using GDELT that reflects the distribution of conflict events in Afghanistan over time.



Posted in **Uncategorized**

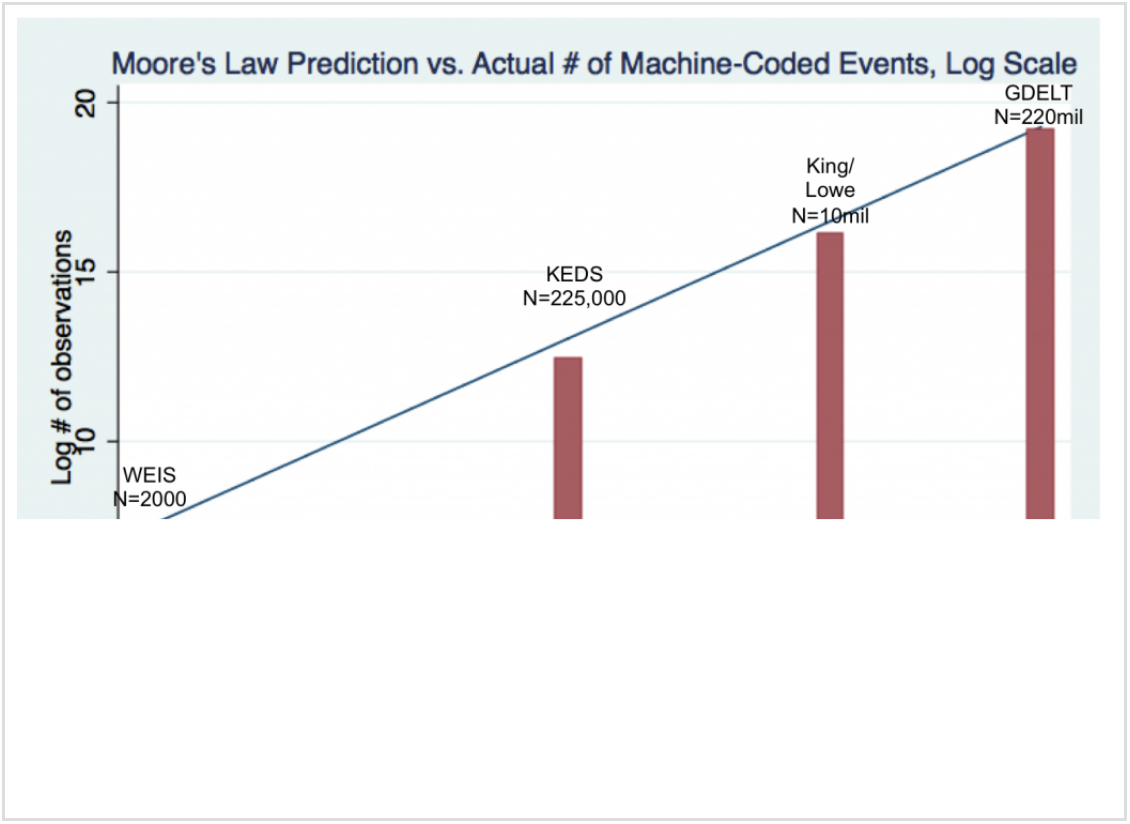
Moore's Law and Event Data

Posted on **March 27, 2013**

In 1965, Gordon Moore predicted that the number of transistors on integrated circuits would double every 2 years. By 1970, the term "Moore's Law" was coined. Since then, Moore's Law has proven shockingly accurate in not only its intended domain (transistors on chips) but across a number of other areas, such as hard disk storage and pixels (see Wikipedia and Microsoft's take). Recently, I found that Moore's Law is also applicable to the number of observations in the largest political event data datasets. Below are the key milestones in political event data. Note that the size of WEIS is a general estimation since the original dataset no longer exists.

- 1978 – World Event Interaction Survey (WEIS): 2,000 observations
- 1996 – Kansas Event Data Set (KEDS): 225,000 observations
- 2004 – 10 Million International Dyadic Event dataset: 10,000,000 observations
- 2012 – Global Database of Events, Location, and Tone (GDELT): 220,000,000 observations

Below, these true values are graphed against what Moore's Law would predict, knowing only that that largest dataset in 1978 was ~1,800 observations. For visual appeal, I plot using a logarithmic scale.



Although the accuracy of Moore's law in the above graph is incredible, what is even more interesting is what this suggests about the future. If the trend holds, the largest political event data dataset should be approaching 1 billion observations around 2016/2017.

Posted in **Uncategorized**

"Data-based Computational Approaches to Forecasting Political Violence"

Posted on **January 10, 2013**

Phil Schrodt, myself, and Benjamin Bagozzi recently had our article "Data-based Computational Approaches to Forecasting Political Violence" published in the Handbook of Computation Approaches to Counterterroism. If the wordcloud seems interesting, check out the article.

Posted in **Uncategorized**

A (slightly) better way to evaluate out-of-sample performance on TSCS data

Posted on **January 3, 2013**

The international relations conflict literature is dominated by time-series cross sections datasets with a binary dependent variable (henceforth referred to as BTSCS). The majority of studies using BTSCS data use a logit/probit model to draw inferences on one or a few "key" variables of interest or compare models. Although the most empirically justified way to actually test which model is better or the importance of a certain "key" variable through evaluating out-of-sample performance, alarmingly few scholars using BTSCS data actually do. Beck et. al. (2000) is a primary exception and current "best practice" approach. Here is a quick recap of their approach:

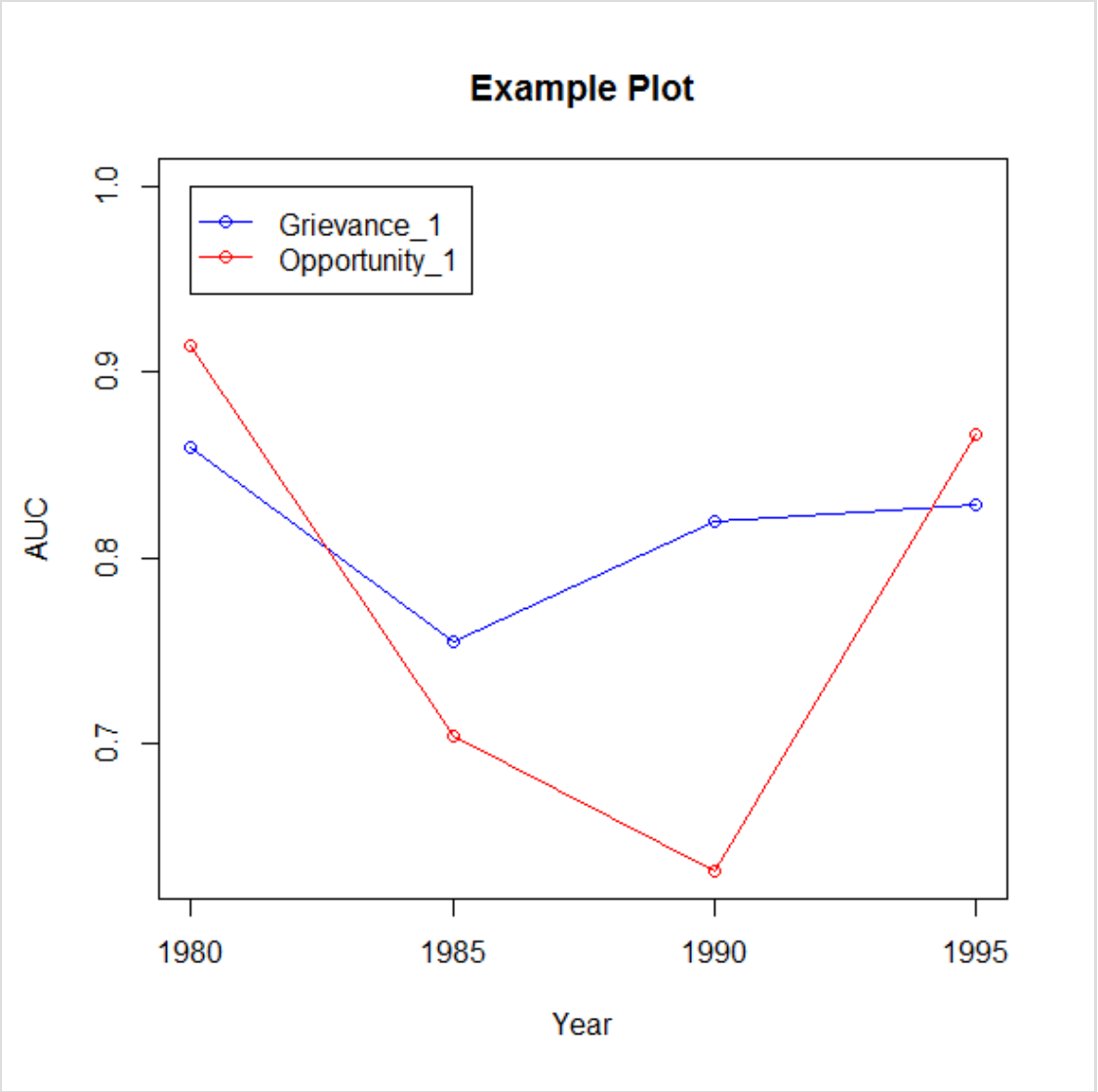
Using the Tucker (1997) dataset (BTSCS at the dyad-year from 1947 to 1989), Beck et. al. set 1960-1985 as the in-sample data and use 1986-1989 as the out-of-sample data to evaluative model performance. This is not a *bad* approach and is infinitely better than purely relying on in-sample metrics. However, this approach means that predictions for 1989 are based on a training model that does not include data from 1986, 1987, and 1988. Thus, potentially valuable data is unnecessarily omitted.

With this in mind, I suggest an alternative "rolling" approach, which iteratively expands the training set by one temporal unit in a way that is conceptually similar to the idea of "online learning" in the machine learning world. The benefits of the rolling approach are largely two-fold. First, unlike Beck et. al's method, the rolling model uses as much data as possible for each forecast. Thus, whereas the Beck approach attempts to generate forecasts for 1989 using training data from just 1947 to 1985, my approach incorporates information from 1986, 1987, and 1988. Second, since performance scores are calculated for each year rather than just on one chunk of years (i.e. for 1986, 1987, 1988, and 1999 individually, and opposed to just 1986-1999 cumulatively), researchers have more information to use when comparing models.

To facilitate discussion, assume we have a TSCS dataset at the state-year level with 100 years of data. Below is the improved, "rolling" approach:

1. Sort the TSCS data ascending by time.
2. Generate count variable for each unique temporal unit.
3. Set the first 75 observations as the training set and the 76^{th} observation as the test set. Depending on the characteristics of your dataset, you may wish to start use anywhere from 50-80% of the data for initial training, instead of the recommended 75%.
4. Estimate and store coefficients using the training set that maximize the likelihood function.
5. Apply these coefficients to the variables from the 76^{th} year to generate predicted probabilities and store measures of predictive accuracy.
6. Repeat steps 3, 4, and 5 by yearly increments until we reach the 100^{th} observation. i.e. retrain the model using the first 76 observations, store coefficients, then generate predicted probabilities for the 77^{th} observation.

Below is some simple code in R that intuitively outlines how this works. The code calculates and plots the out-of-sample AUC scores for the Grievance and Opportunity models in of Collier and Hoeffler (2004) (click here for the data):



The plow above shows that the “rolling” approach generates an AUC score for each time period, as opposed to a single, cumulative score for 1980 to 1995. In this example, neither the Grievance nor Opportunity model appears to outperform the other, since their AUC scores overlap.

```
1   ############################
2   ##code for 1 unit ahead model evaluation of BTSCS data
3   ##James E. Yonamine
4   ##1/3/2013
5   ##data from http://www.apsanet.org/content_29436.cfm
6   ##run time: 12 seconds
7   ############################
8   rm(list=ls())
9   library(foreign)
10  library(ROCR)
11  library(lattice)
12  library(Zelig)
13  library(Amelia)
14  library(stats)
15
    data <- read.dta("collier and hoeffler 2004.dta")
```

```
16 data<-data[complete.cases(data$warsa),]  #this drops all rows
17 a.out <- amelia(data, m = 1, ts = "year", cs = "country")
18 data<-a.out$imputations[[1]]
19 data<-data[order(data$year),]
20 data<-cbind(data, data.frame(count=cumsum(c(TRUE,data$year[-1]
21
22 num<-data[nrow(data),ncol(data)]  ##figure out how many unique
23 start = 4 ##this determine what portion of the data to use as
24 step=1   ##how many steps ahead do you want to forecast
25 history <- matrix(data=0, nrow=(num-start), ncol=3)
26 colnames(history) <- c("year", "griev_1.auc", "opp_1.auc")
27
28 for (i in start:(num-1)) {
29     set.seed(i)
30     train.data<- data[which(data[,ncol(data)] <= i),]  #set i
31     test.data <- data[which(data[,ncol(data)] ==(i+step)),] #
32
33     #########Grievance################
34     grievance.1 <- glm(warsa ~  elfo + rf + pol16 + etdo4590 +
35     grievance.1.predict <- predict(grievance.1, newdata=test.d
36     grievance.1.y.hat<-as.matrix(grievance.1.predict)
37
38     ## Establish prediction objects from ROCR package
39
40     y<-as.matrix(test.data$warsa)
41     with.predict <-prediction(grievance.1.y.hat,y)
42
43     ## Calculate and store the AUC
44     with.auc <- performance(with.predict, measure = "auc")
45     history[(i-start)+1,2] <- as.numeric(unlist(slot(with.auc,
46
47     #########Opportunity###############
48
49     opportunity.1 <- glm(warsa ~sxp + sxp2 + coldwar + secm +
50     opportunity.1.predict <- predict(opportunity.1, newdata=te
51     opportunity.1.y.hat<-as.matrix(opportunity.1.predict)
52
53     ## Establish prediction objects from ROCR package
54     y<-as.matrix(test.data$warsa)
55     without.predict <-prediction(opportunity.1.y.hat,y)
56
57     ## Calculate and store the AUC
58     without.auc <- performance(without.predict, measure = "auc
59     history[(i-start)+1,3] <- as.numeric(unlist(slot(without.a
60
61     ### store the temporal component
62     history[(i-start)+1,1] <-test.data[1,4]
63 }
64
65 #####plot it####
```

```
62
63  history<-as.data.frame(history)
64  ts(history$year)
65  ymax=1
66  ymin=min(history)
67  yrange=range(ymin, ymax)
68  plot(history[,2], type="o", col="blue", xaxt="n", ylim=yrange,
69  lines(history[,3], type="o", col="red")
70  axis(1, 1:nrow(history), lab=history[1:4,1])
71  legend(1, c("Grievance_1","Opportunity_1"),
72     col=c("blue","red"), pch=21, lty=1)
73
74
75
76
77
```

gistfile1.r hosted with ♥ by GitHub view raw

Posted in **Uncategorized**

What can Kimbo Slice teach us about predictive models?

Posted on **December 13, 2012**

— In the backyards of Miami, my money is on Bueno de Mesquita, every time.

I have tried very hard to keep my website/blog/twitter focused solely on my professional interests – using open source data to forecast socially driven outcomes. Now, I'm going to bend the rules a bit and bring in my love of combat sports in order to apply lessons from mixed martial arts (MMA, or "cage fighting" or "ultimate fighting" to the layperson) to forecasting models.

Last Saturday, Fox aired "UFC on Fox 5", which was undoubtedly the greatest collection of MMA talent ever aired on a free TV. The main event – Nate Diaz vs. Benson Henderson – garnered 5.7 million views. Not bad, but consider this: Kimbo Slice has fought 3 times on free TV, each time surpassing 6 million views (6.1, 6.45, and 7.1 million to be exact). Until a string of losses that exposed him as a D level fighter, Kimbo was among the biggest draws in all of combat sports. But why? I believe that it derives from our obsession with the mythical. In the context of fighting, we seem to be drawn to someone who, for untraditional reasons, seems to be invincible. In terms of MMA, the fighters who achieve mythical status share two things in common: 1) untraditional or secret training methods and 2) a string of dominant victories over easy opponents. As I argue a bit later, the same is true of predictive models

In the early stages of MMA, the Gracie family dominated. For many years, their brand of jiu-jitsu (submission fighting on the ground) allowed physically inferior men like Royce to appear invincible in actual no holds barred fighting. Their highly technical method of ground fighting seemed like magic to the untrained eye, and since virtually no one in the United States was familiar with jiu-jitsu at the time, it just seemed like magic. Eventually, the Gracie's began to fight better competition; Kazushi Sakuraba tarnished the Gracie legacy by beating Royce, Renzo, Royler, and Ryan, and Matt Hughes drove the final nail in the coffin when he mauled Royce.

The Russian fighter Fedor Emelianenko regained and arguably surpassed a Gracie-level of mystique. Like the Gracie's, the specific details of his training regimen were not know outside of Russia, and the only glimpses of his training U.S. audiences saw were Youtube clips of minimalist workouts on Russian playgrounds. He was (and likely still is) a secretive man who seemingly cared more about his religious devotion than fighting (his Orthodox priest often accompanied him to fights). Also, like the Gracies, Emelianenko gained mythical status by beating a mix of legitimate, semi-legitimate, and absurdly illegitimate competition.

Kimbo Slice achieved event greater U.S. fame than the Gracie's or Emelianenko through a series of backyard street fights in Miami that circulated on YouTube. He destroyed lesser competition (i.e. people with no idea how to fight) in dramatic fashion (i.e. he once popped out someone's eyeball. Seriously.). No one really knew his fighting background or training, but people seemed to assume that he was so innately tough that it didn't much matter. By the time he fought James Thomson live on CBS in 2008, he had reached mythical status. Of course, this came crashing down when he was knocked out in 14 seconds by unranked journeyman Seth Petruzelli.

So what does this all have to do with forecasting models? As with MMA fighters, we seem

to be intuitively drawn to black box, secret models. Additionally, like in the fight game, models can artificially inflate their reputation by "beating up" on easy problems.

Like with out fighters, we seem to be intuitively drawn towards secret, mythical approaches to prediction. Just consider the absurdity of the history of human forecasting: Oracles interpreting hallucinogenic dreams, astronomers staring at the heavens, monks pouring over ancient texts, etc.. My sense is that many still seem to think that a secret formula exists that can predict the world. Many practitioners, like Bruce Bueno de Mesquita or the CIA, certainly help to propagate this myth (see the History Channel special that pitted Nostradamus vs. BDM). I suspect if you surveyed 1,000 random people about whether academics using open source data and open-source statistical packages could build better forecasts of civil unrest than the CIA using only classified data and proprietary algorithms, most would pick the CIA. I'd bet anything on the open source team. In predictive models as in fighting, there is no secret approach to success.

Additionally, similar to the way in which fighters achieve mythical status by often padding their records fighting easy competition, predictive models have a few tricks to "pad" their records as well. First, and a favorite of the game theorists, is to use a model to make non-falsifiable predictions. Thus, no matter what happens, the game theorists can make a strong case that his or her model correctly predicted it after the fact. Second, the primary tool of empirically driven prediction models is to inflate results by predicting that tomorrow will look a lot like today. For outcomes like whether or not a country will be at civil war tomorrow, this approach is correct 90+% of the time.

In reality, neither mythical fighters nor mythical predictive models exist. Everyone knows exactly how the best fighters — Jon Jones, Anderson Silva, Georges St. Pierre — train. These fighters get to that level by having slightly more genetic gifts, work slightly harder, and having slightly better coaches, all of which contribute to a minor advantage over their opponents come fight time. This is almost identical to the predictive model world. In actual, objective tests of predictive accuracy (such as Kaggle competitions), the wining models tend to only narrowly beat the competition and the methodological approaches tend to be highly similar. The reality of predictive models is just like that of fighting: there are no mythical approaches. The winning teams tend to put in slightly more time on slightly more powerful machines with teams comprised of slightly more experienced modelers.

With fighters as with predictive models, if it seems to good to be true, it almost certainly is.

Posted in **Uncategorized**

---

3 things to pay attention to when

analyzing predictive accuracy

Posted on **November 9, 2012**

{{insert obligatory Nate Silver reference to connect the content of this post to current events}}

As is readily obvious from the content of this blog, I think and write frequently about predictions, and I constantly advocate that the only way to test whether a person or model (of the non-person type, unfortunately) actually helps us better "understand" the world is to evaluate how well he/she/it can predict. Despite the emphasis of predictive accuracy, there are a number of complications to keep in mind when evaluating a model's predictive accuracy that tend to be overlooked. Here are three that I believe to be particularly important:

First, and perhaps most importantly, new models of the world that ultimately prove correct occasionally generate weaker initial predictive accuracy than long established models that ultimately prove poor reflections of reality. Consider the debate between proponents of the geocentric vs. heliocentric universe. In western scientific history, geocentric models predated heliocentric alternatives, giving their (geocentric) proponents more time for model refinement. This meant that geocentric models occasionally generated more accurate predictions than newer heliocentric models, even though we now know with certainty that the sun is the center of our solar system. Thus, scientific progress often requires one step backwards in terms of predictive accuracy to ultimate take many steps forward (I borrow this point from Manzi).

Second, one of the major problems with reliance on in-sample testing is the possibility of overfitting: it is impossible to know if purported statistically significant covariates are simply fitting error or a true relationship. Especially in empirical studies of conflict, inferences drawn from purely in-sample models simply cannot be trusted. Rather, the gold standard for evaluating a model's performance is out-of-sample (but don't take it from me, see Beck, King, and Zheng 2000). In a perfect world, this would entail training a model on a dataset, then making predictions as we collect new data in real-time. In practice, this can be difficult so we simulate this process by separating our data into an in-sample (often called training) and out-of-sample (often called validation or test) set. It is critical to note that it is still possible to overfit a model using this out-of-sample set up. This can occur when researchers do the following:

1. train a model on the in-sample data
2. evaluate its predictive performance on the out-of-sample set
3. change the original models based on the results
4. repeat this process until results become satisfactory

There are many techniques to avoid overfitting in an out-of-sample framework. One of the

most straightforward and commonly employed is to divide a dataset into three parts: training, validation, test. To avoid artificially enhancing predictive through overfitting, simply add a fifth step:

5. pick a single model and tests its predictive accuracy on the test set

It is critical that the test set must only be used one time. Note that various other iterative sampling approaches also exist to avoid overfitting.

Third, it is always possible that a model's high degree of predictive accuracy is due entirely to luck. To crudely illustrate this, I borrow an example that Warren Buffett to illustrate the role of luck in stock picking. Imagine there was a NCAA basketball style, rock-paper-scissors single elimination tournament with ~1 billion people, with the winner getting paid $1million. As each round progressed, the excitement would build. After round 28, we would be down to 4 individuals. By this point, they would be making the talk show rounds, likely describing their secret formula for success. MSNBC would eat it up. Finally, a winner emerges, he/she probably lands a book deal with a title like "Get Rich Now: How my approach to paper rock scissors can make you millions". Meanwhile, the entire thing was luck.

With the number of people building predictive models (especially in finance), odds are than some models will achieve spectacular runs of success based entirely on luck. Others may achieve success because their model is actually superior. Differentiating between a dominant model and pure luck can be very difficult, especially when the outcome-of-interest is observed infrequently (like in elections).

Posted in **Uncategorized**

What is data science?

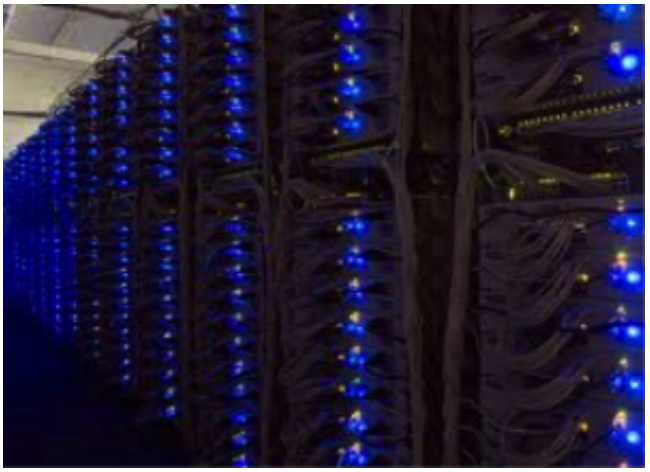
Posted on **October 15, 2012**

"Data science" is a new hot phrase that has largely coincided with the rise of "Big Data". We often use the term "data science" loosely to mean "things that we do with big data", and while this description is accurate in the big picture sense, it can also be overly vague. In some ways, definitional vagueness might not be such a bad thing. For example, maintaining a broad definition of "data science" may motivate students to build a broad skill set and facilitate more interdisciplinary collaboration. On the other hand, more nuanced terminology could assist recruiters better target candidates with specific skill sets, and help educational institutions better structure current/future "data science" curriculum. Regardless, I'm certainly not the first to try to unpack "data science" and have no illusions that I am the first to suggest (or blog) that "data science" contains three main component

parts. So, if you have already seen this, I apologize and claim no intellectual ownership. If this is new, I hope you find it insightful.

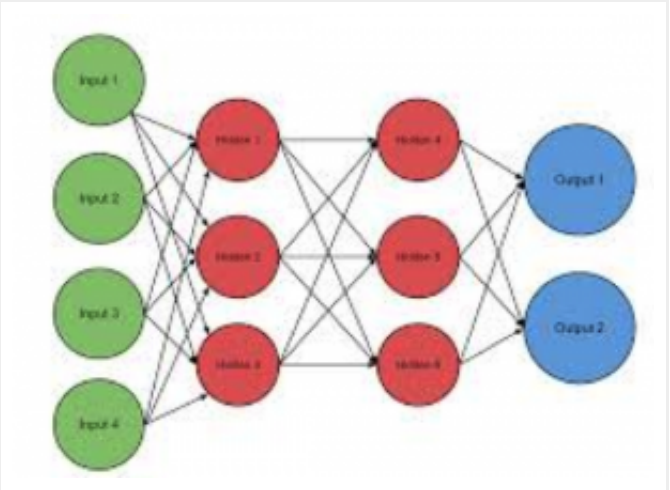
I think concept of “data science” is most effectively understood by breaking it down into its three main components:

1. data storage and management
2. data analytics
3. data visualization



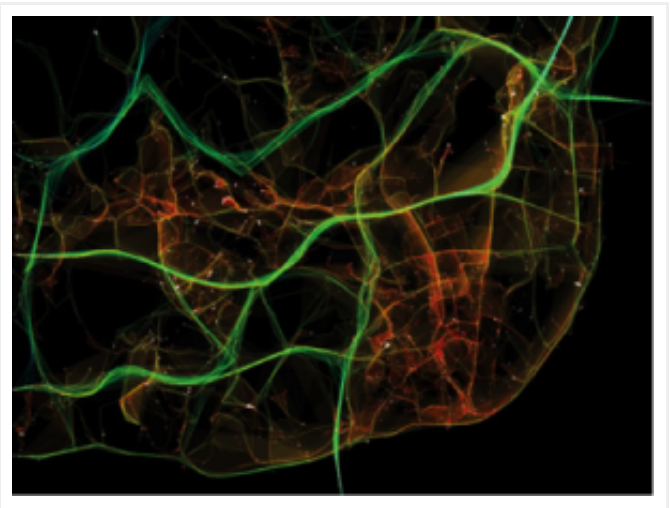
— (server)

Data storage and management: Electronic data is becoming immense. And not just in the “wow, this .csv is too big to read into R”, but also in the “we need a hundred acre server farm to physically store our data”. I refer to *data storage and management* as the ability to build systems able to both transfer new data in real time to storage servers as well as efficiently retrieve data from storage, though one could take this a step further and lump in the physical engineering aspects of building and maintaining servers. These tasks require a deeper understand of the how computers operate and skills most likely learned computer science/computer engineering/information science and technology. Key software include (my)Sql, MySql, MongoDB, MapReduce, Hadoop.



— (neural network framework)

Data analytics: Data analytics is the process of extracting meaningful patterns from the data that lead to accurate, replicable, and non-obvious predictions. Though the emphasis on predictive accuracy may seem overly restrictive, I'd argue that it isn't since most actionable data analytics findings are inherently predictive. For example, data analytics focusing on outcomes ranging from increase ad clicks to effective cancer treatments to streamlining car traffic are all predictive: a model predicts which ads someone is more likely to click, which cancer treatment is most likely to be successful, and which system of traffic like is most likely to reduce traffic. In most contexts, *data analytics* is becoming nearly synonymous with "data mining", which happens to be synonymous with "machine learning". Effective data analytics requires a strong understanding of computational statistics and knowledge of the canonical machine learning/data mining algorithms (i.e. SVMs) and tools (i.e. boosting) likely obtained by joint training in statistics and computer science departments. Key software include R, Python, Matlab, Weka, and a number of specialized and proprietary data mining programs. Check out my the book "Data Mining with R" for a great introduction to…data mining with R.



— (traffic flow in Lisbon, via http://www.visualcomplexity.com/vc/project_details.cfm?

id=728&index=728&domain=)

Data visualization: Visualizing data can serve a number of purposes. At the extreme, data can be visualized for purely aesthetic purpose – I'd guess most people are buying Manuel Lima' Visual Complexity as a coffee table book of art, not as a methodology textbook (by the way, if you haven't seen it, check it out, it's incredible). However, the goal of most data visualizations is to help humans better process large amount of information. Of the three component parts of "data science", it is the most difficult to measure "goodness" when it comes to data visualization. Whereas it is relatively straightforward to test which server is more energy efficient or which machine learning algorithm is reducing predictive error, determining which visualization is most effectively conveying complex information tends to be harder since it is a more difficult concept to quantify (suggestions for evaluating data visualization will be a future post). Good data visualization requires a challenging combination of technical computer science skills and artistic aptitude. The social science community tends to rely on Python and R libraries, though LOTS of other visualization software exists. Check out www.flowingdata.com and http://junkcharts.typepad.com for some very cool big data visualizations.

Posted in **Uncategorized**