

Lewis-Beck and Tien House Forecast Model

Harry Bendekgey

July 17, 2018

Michael Lewis-Beck and Charles Tien propose a simple model, called the Structure-X model, to forecast house elections. The authors mix a structural model and an expert model to produce their forecast, hence the name.

Structural

I will start by analysing the structural portion of the model:

```
summary(fit1)$coef
```

##		Estimate	Std. Error	t value	Pr(> t)
##	(Intercept)	-37.6600899	10.7199952	-3.513070	1.383682e-03
##	midterm	-21.5249085	5.7490993	-3.744049	7.399807e-04
##	rdi_growth	9.6747993	2.0517143	4.715471	4.847174e-05
##	pres_app	0.4626148	0.2209676	2.093587	4.457067e-02

In this model, the change in incumbent party seats is predicted by three variables: the percent change in Real Disposable Income (RDI) in the first two quarters of the election year, the approval rating of the President in June of the election year, and whether or not it is a midterm election. Each of these predictors is rooted in theory and point in the directions we'd expect: good economic growth is associated with pro-incumbent sentiment, as does Presidential approval. Incumbents tend to lose seats during midterm years, having done so in every election since 1948 other than '98 and '02 (the latter being an exceptional case as the election immediately following 9/11)

It is surprising that Presidential approval is barely significant. This is because of 1948. We are regressing on all elections since 1948, and a single election can have huge influence on the model if it has a high residual, or high leverage. A high residual means that the result is unexpected given the linear model. 1948 featured a 75 seat gain for the incumbent party under mediocre Presidential approval numbers albeit a booming economy. We have never seen a wave that huge since. 1948 is also a high leverage year, meaning that the parameters are unlike any other election year since. It has the highest RDI growth of any election year since, so it drags up the coefficient and significance of the RDI predictor.

Normally we must be careful removing data points because we don't like them. But the choice to run regression on election years since 1946, which is the standard in the literature, is somewhat arbitrary. I did not include 1946 in this model because I could not find RDI data for that year. In addition, I do not think the results of 1948 are particularly predicting of 2018; a lot has changed in that time, and I believe we risk more by giving 1948 huge influence over 2018's results than by running regression on elections starting in 2018. Removing 1948, we get:

```
summary(app_fit)$coef
```

##		Estimate	Std. Error	t value	Pr(> t)
----	--	----------	------------	---------	----------

```
## (Intercept) -44.8264209  8.9445895 -5.011568  2.254802e-05
## midterm    -22.2080193  4.7048670 -4.720223  5.127921e-05
## rdi_growth   6.1740641  1.8881857  3.269839  2.703265e-03
## pres_app     0.6944504  0.1895943  3.662824  9.556705e-04
```

We can see that RDI growth as a predictor lost a third of its power, just by removing 1948. In contrast, Presidential approval gained 50% more power.

```
summary(app_fit)$r.square

## [1] 0.7211392

summary(dis_fit)$r.square

## [1] 0.7268682

summary(net_fit)$r.square

## [1] 0.7272933

summary(share_fit)$r.square

## [1] 0.7266794
```

At this point, I want to further note that instead of using Presidential approval, you could use Presidential disapproval, net Presidential approval, or Presidential approval share (approval scaled to remove non-respondents) and end up with almost identical predictive power.

We now regress, using Gallup's June average of 42 Presidential approval, 52 disapproval, and an estimate for RDI growth (until the June data is released):

```
params <- data.frame(rdi_growth=1.47, pres_dis = 52, midterm = 1, pres_app=42, pres_net=-10, pres_share
app_interval <- predict.lm(app_fit, params, interval = "prediction")
dis_interval <- predict.lm(dis_fit, params, interval = "prediction")
net_interval <- predict.lm(net_fit, params, interval = "prediction")
share_interval <- predict.lm(share_fit, params, interval = "prediction")
```

```
app_interval

##          fit          lwr          upr
## 1 -28.79165 -55.96634 -1.616958

dis_interval

##          fit          lwr          upr
## 1 -31.47286 -58.61259 -4.333116

net_interval

##          fit          lwr          upr
## 1 -30.40019 -57.40427 -3.3961

share_interval

##          fit          lwr          upr
## 1 -30.23646 -57.25686 -3.216062
```

This is somewhat unsurprising: in an age of rising partisanship, Trump's low-ish approval ratings don't look as bad for him as his high disapproval ratings. This is because fewer and fewer people are responding undecided, causing Trump's approval and disapproval to look high if compared to a theoretical comparable candidate in the 1960s. thus I will use poll share to correct for this (approval respondents divided by all decided respondents, measured in percentage point deviation from 50)

The other thing to note is that this confidence interval is shockingly close to Abramowitz's. Instead of a point estimate of Republicans losing 33 seats, we estimate them to lose 30 seats. The range of the interval is tighter by about 19 seats, but with only a couple dozen races analyzed it's hard to know what the true uncertainty of a model is. This similarity is especially powerful when you realize that these two models regress on totally different measures: one uses Presidential approval and economic conditions, while the other uses generic ballot results an dcurent congressional distribution.

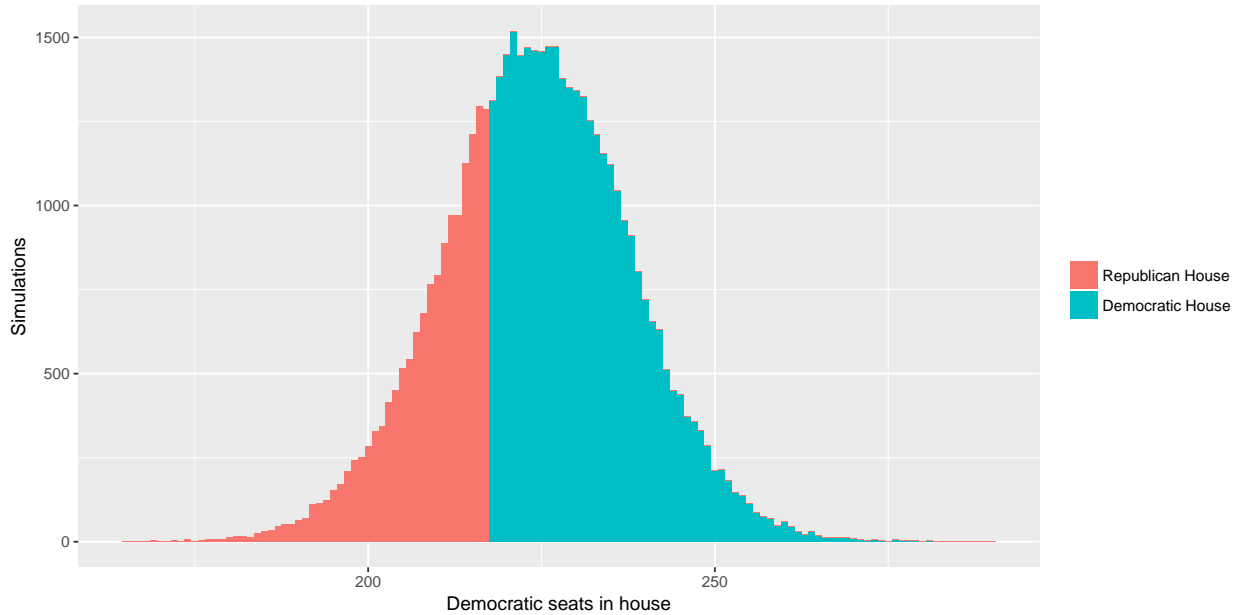
```
interval <- predict.lm(share_fit, params, interval = "prediction", se.fit = TRUE)
t.cutoff <- qt(0.975, df=interval$df)
se <- (interval$fit[1] - interval$fit[2])/t.cutoff
dem_win_pct <- pt((-24-interval$fit[1])/se,df=interval$df)
```

```
se
## [1] 13.23056

dem_win_pct
## [1] 0.679606
```

Thus we give Democrats about a two-thirds probability of taking the house with the following distribution:

```
chrseats <- rt(50000, interval$df) * se + interval$fit[1]
dseats <- round(194 - chrseats)
forecast_df <- data.frame(dseats)
ggplot(aes(x=dseats, fill=(dseats > 217)), data=forecast_df) +
  geom_histogram(binwidth=1) + xlab("Democratic seats in house") +
  ylab("Simulations") + scale_fill_discrete(name=element_blank(),
    breaks=c("FALSE", "TRUE"),
    labels=c("Republican House", "Democratic House"))
```



Expert

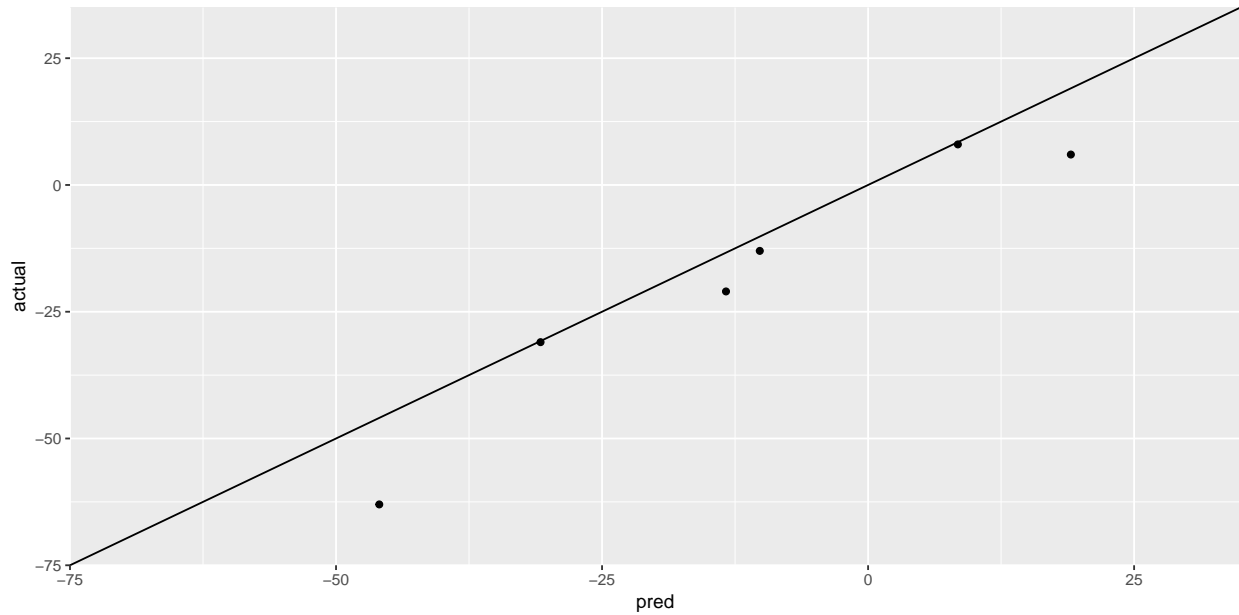
One problem with this model is that it does not take into account current control of the house. Exposure models tell us that knowing how many seats incumbents currently have help us predict how many they risk to lose. The structural model above has all the properties we want from a model, but there is predictive power missing; about 30% of variability still remains unexplained. To close this hole, Lewis-Beck and Tien enlist the help of an expert model.

An expert model tries to predict net seat change using seat ratings from organizations like Cook, Sabato, or Rothenberg. Individual house and senate races are marked as safe, solid, likely, lean, or tossups for one party or the other. Those ratings are made to predict the outcome of individual races, and so we ask what the relationship is between how seats are rated and what the net seat change is.

Lewis-Beck and Tien make use of Rothenberg's seats-in-play differential: the difference between the number of out-party seats considered competitive, minus the number of in-party seats considered competitive. If all competitive seats are split (i.e. each party wins half of them) then you would expect the net change to be half of the seats-in-play differential. In actuality, the whole differential ends up being a good predictor of net seat change, meaning that when Rothenberg lists more seats for one party, most of those seats end up getting taken.

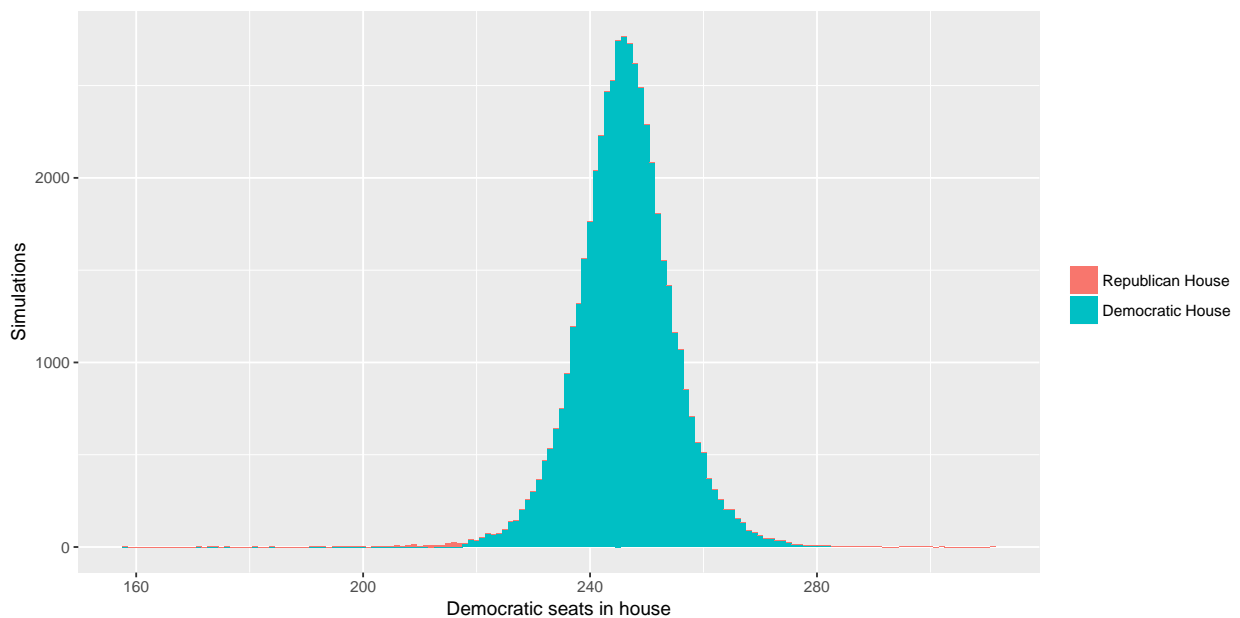
In 2006 the Rothenberg differential was -31. Republicans lost 31 seats. In 2008, -16 was Rothenberg's differential for a net change of -21. In 2010 a differential of -64 predicted a swing of -63 seats. But this metric is not perfect. In 2014 and 2016, Rothenberg's differential was 2 and 46 in favor of Democrats, with actual swings -13 and 6.

The authors combine these two metrics by averaging them. Rothenberg has only been producing ratings since 2006, so we have very few data points, but this method of combining them greatly reduces the error from either individual prediction. We see:



We notice that the predictions seem to be systematically biased in favor of incumbents, having overestimated incumbent performance for every single election since 2006. The size of this bias is on average 7.5 seats. If we make this adjustment, calculate standard error, and produce a distribution of seats, we see:

```
chrseats <- rt(50000, 6) * sigma + mu
dseats <- round(194 - chrseats)
forecast_df <- data.frame(dseats)
ggplot(aes(x=dseats, fill=(dseats > 217)), data=forecast_df) +
  geom_histogram(binwidth=1) + xlab("Democratic seats in house") +
  ylab("Simulations") + scale_fill_discrete(name=element_blank(),
    breaks=c("FALSE", "TRUE"),
    labels=c("Republican House", "Democratic House"))
```



This is a very different picture. The reason for this is Rothenberg's seats-in-play differential of -59, almost as big as 2006's. That year, Rothenberg predicted the swing much better than the structural model, but in 2018, it predicted a huge swing of 48 and saw only 6 seats swing. It's important to be careful who you trust, and decide for yourself if pundits are doing better in 2018 than in 2016.

Adding the X-part of the Structure-X model pulls the seat change further towards Democrats and reduces the standard error, causing a great deal of confidence. The maximum error of the adjusted Structure-X model in the past 6 elections is 10 seats.

```
mu
## [1] -52.16323
mean(dseats > 217)
## [1] 0.99676
```

Thus we predict Republicans will lose 52 seats, and estimate a 99.7% chance of Democrats taking the house.