Election Forecasting Dynamics

https://twitter.com/nntaleb/status/794608806072352769

Election Forecasting with Semi – Continuous Updating

A Dynamic View of Forecasting

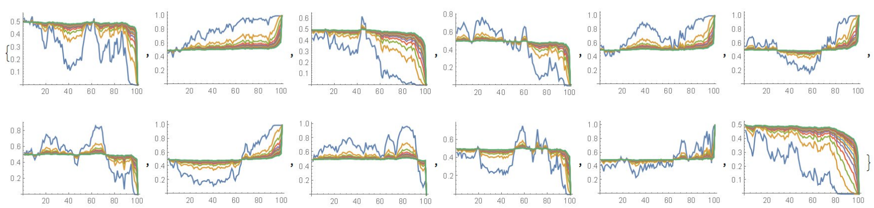
What makes a good forecaster? As traders we know that the final outcome is just a piece of the pie. Every day's P/L matters. You need to consider the steps in the process. In fact you can tell a bad forecaster before the end event, and tell **when** you can pronounce forecaster A better than forecaster B. In the real world, a forecaster who is also a market maker **can go bankrupt** before final outcome.
Also this shows how it is worse to **produce no change in forecast** than keep changing, and how to calibrate changes to volatility.
The math is as follows. Let b_{t_0} be your "price" $\in[0,1]$ time t_0, your "probability", and $b_{t+\Delta t}$ your price time $t+\Delta t$, etc. Assume elections happen time τ.

Since your forecast is left hanging, you are evaluated at *how little opportunity one can arbitrage you*, that is buy from you at b_{t_0} and sell at $b_{t_0+\Delta t}$. Hence your quality of forecasting is some norm $\|b_{t_0}-b_{t_0+\Delta t}\|$. This reduces to the Brier metric with would be $\|b_{t_0}-b_\tau\|$, $b_\tau \epsilon\{0,1\}$ being the final result. The Brier metric uses Norm L2 (squared deviations) but your P/L is is norm L1 (absolute deviations) so the latter is preferable.

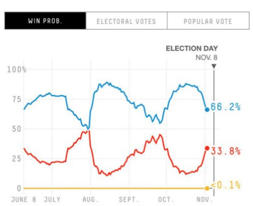
$$\mu(t_0,\ \tau,\ \Delta t) := \sum_{i=0}^{\frac{\tau-t_0}{\Delta t}} \left\| b_{\Delta t(i+1)+t_0} - b_{i\Delta t+t_0} \right\| \tag{1}$$

The probability can see that as $\Delta t \to 0$ we have a nonanticipating Ito integral.

As we see in the graph below, the best forecaster is (under metrics $\|_1$ and $\|_2$) the one who is is minimally wrong throughout, that is the half way line. The worst is the blue one. You can already tell half way that regardless of final outcome, the blue is the worst forecaster.

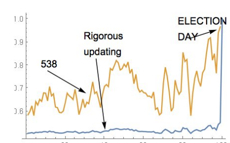


In other words, it is how you can best forecast yourself. Apply that to today's events. Compare to 538. These people have a negative P/L regardless of if they guess the election right. They are already bust.



Tutorial on How to Price Elections with Updated Point in Time Estimation

Assume W is a continuous state variable determining the final result.



Let us start the model from the very basics. Very very basics of stochastic calculus. We have the election estimate *F* a function of a state variable *W*, a Wiener process WLOG. *W* can be an estimate, or some other variable. the estimation error can be integrated into the variance of *W*. *W* has for simple dynamics (arithmetic B M, we can transform later):

$$dW = dt\,\mu + dZ\,\sigma \tag{2}$$

By Ito' s Lemma:

$$dF = \frac{\partial F}{\partial t}\,dt + \frac{\partial F}{\partial W}\,dW + \frac{1}{2}\,\frac{\partial^2 F}{\partial W^2}\,dW \tag{3}$$

Ito's calculus allows dt^2 and $dt\,dW$ to vanish. The idea of no arbitrage is that a continuously made forecast must itself be a martingale of sorts. Apply the Black Scholes (or a standard no arbitrage) argument; Replacing with (2), and assume μ=0 to simplify WLOG

$$dF - \frac{\partial F}{\partial W}\,dW = 0 \tag{4}$$

We end up with the partial differential equation:

$$\frac{\partial F}{\partial t}\,dt = -\frac{1}{2}\,\frac{\partial^2 F}{\partial W^2}\,\sigma^2 \tag{5}$$

which is, basically, the heat equation. We have for terminal conditions: $F|_{t\to 0} = \theta[W]$ where θ is the Heaviside Theta function. We can try to solve on Mathematica (by fudging, inverting the backward-forward equation)

```
Eq = D[F[W, t], t] == 1/2 σ^2 D[F[W, t], {W, 2}];
tc = F[W, 0] == HeavisideTheta[W];
sol = DSolve[{Eq, tc }, F[W, t], {W, t}]
```

$$\left\{\left\{F[W,\ t] \to \frac{1}{2}\left(1 + \mathrm{Erf}\left[\frac{W}{\sqrt{2}\ \sqrt{t}\ \mathrm{Abs}[\sigma]}\right]\right)\right\}\right\}$$

which is the CDF of a the Normal distribution for P≥W. If W is a "poll", we can transform ϕ^{-1}:$(-\infty,\infty) \to [0,1]$ to get it to translate.

Et finito!

In[ ]:=
```
Plot[Evaluate[Table[(1 / 2) (Tanh[x / k] + 1), {k, 1 / 2, 3, 1 / 2}]], {x, -5, 5}]
```
Out[ ]=