Surprisingly-Popular Forecasting of Range Variables

May 4, 2021

Emile Servan-Schreiber & Maurice Balick

This note describes Hypermind's novel adaptation of the "Surprisingly Popular" algorithm of Prelec, Seung, & McCoy (2017) to crowd-forecasting of variables along a continuous range.

Prelec, Seung, & McCoy (2017) asked subjects to vote for the correct answer among several in a discrete set, and also to predict how others would vote. They prove that the "surprisingly popular" (SP) answer in a set of discrete answers is also the most likely correct answer. They compute a "*prediction-normalized vote*" (V) for each discrete answer and show that the correct answer has the highest V.[1]

In the context of experimental data, they propose several approximations to compute V for each answer i:

$$V(i) = perso(i) \times \sum_{j \neq i}^{j} \frac{predict(j|i)}{predict(i|j)} \qquad \textbf{(1)}$$

Where:

- $perso(i)$ is the percentage of participants who voted for answer i;
- $predict(i|j)$ is the proportion of predictions for answer i among those who voted for answer j.
- approximate **0/0** $\equiv$ **0**

Extension to range variables

In our case the answers are not discrete. Instead, both the answers and the predictions are expressed as probability distributions over a range of values. Additionally, we ask subjects not to

[1] Prelec, Seung, McCoy (2017) *A solution to the single question crowd wisdom problem: Supplementary Information.* Nature, 541.

vote for a correct answer, but to forecast the future value of a continuous target variable. Figure 1 shows how the platform elicits a user's input.

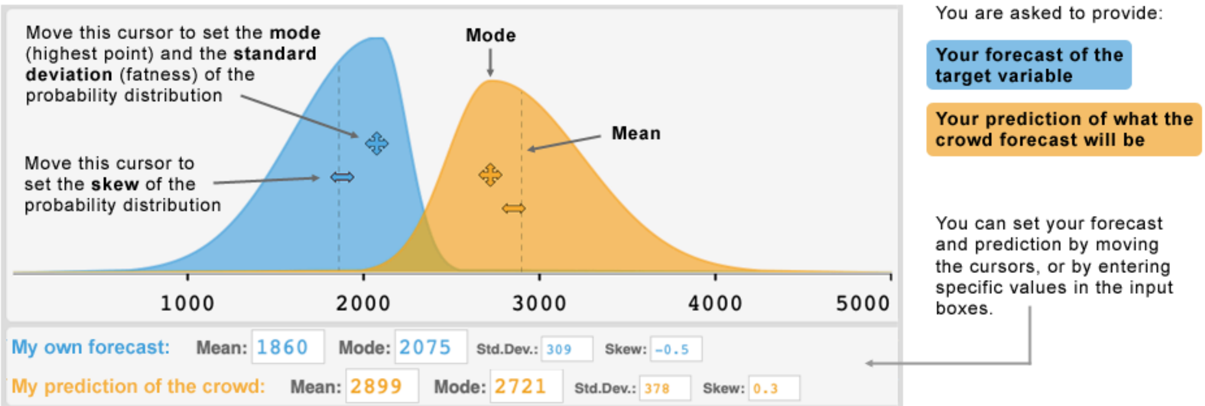

FIGURE 1 *- How Hypermind's Prescience platform elicits from each user both a personal forecast for a continuous variable (x-axis) and a meta forecast, i.e. a prediction of the crowd's forecast. The user can control the mode, the standard deviation and the skew of her distributions, either by entering specific numbers, or by adjusting a couple cursors.*

To adapt the discrete case described by Prelect et al (2017) to our case of continuous forecasting, we first divide the target variable's range (x-axis in Figure 1) into a sequence of hundreds of contiguous bins. A user's forecast and meta-forecast are then translated into vectors of probabilities over all bins.

Then we compute the aggregated "crowd" forecast and meta-forecast as normalized vectors of summed user probabilities in each bin:[2]

- *fcst* = the vector over all bins of the personal probabilities in each bin, normalized;
- *meta* = the vector over all bins of the meta probabilities in each bin, normalized;

Now we compute the prediction-normalized forecast in each bin, using the Prelec et al (2017) formula (1) above:

$$V(i) = fcst(i) \times \sum_{j \neq i}^{j} \frac{meta(j|i)}{meta(i|j)} \qquad \textbf{(2)}$$

Where:

- *fcst(i)* = probability for bin i in the *fcst* vector (e.g., 0.08);
- $meta(i|j)$ = the average meta probability in bin i considering only the users who have a non-zero personal probability in bin j.
- approximate **0/0** ≡ **0**

[2] If user weights are available, perhaps based on past forecasting success, then the aggregations could also be computed as weighted averages.

The vector V of prediction-normalized probabilities over all bins is our version of the "surprisingly popular" forecast distribution over the target variable's range. If we need a point forecast, we just report the mean of that distribution:

$$\text{Surprisingly popular point forecast} = \frac{1}{\sum_i V(i)} \times \sum_i (target(i) \times V(i)) \qquad \textbf{(3)}$$

where $target(\mathrm{i})$ = the target variable's value in the ith bin (e.g., 15% growth, or 1,200 EFLOPS);

Figure 2 shows an actual example of the results of our computations, taken from a recent Hypermind forecasting contest on artificial intelligence progress at the 2030 horizon. The blue curve ("actual crowd") aggregates the personal forecasts, while the orange curve aggregates the meta predictions ("predicted crowd"). The gray curve ("collective intelligence") shows the prediction-normalized probabilities that result from aggregating the surprisingly popular forecasts. The solid vertical line indicates the surprisingly popular point forecast.

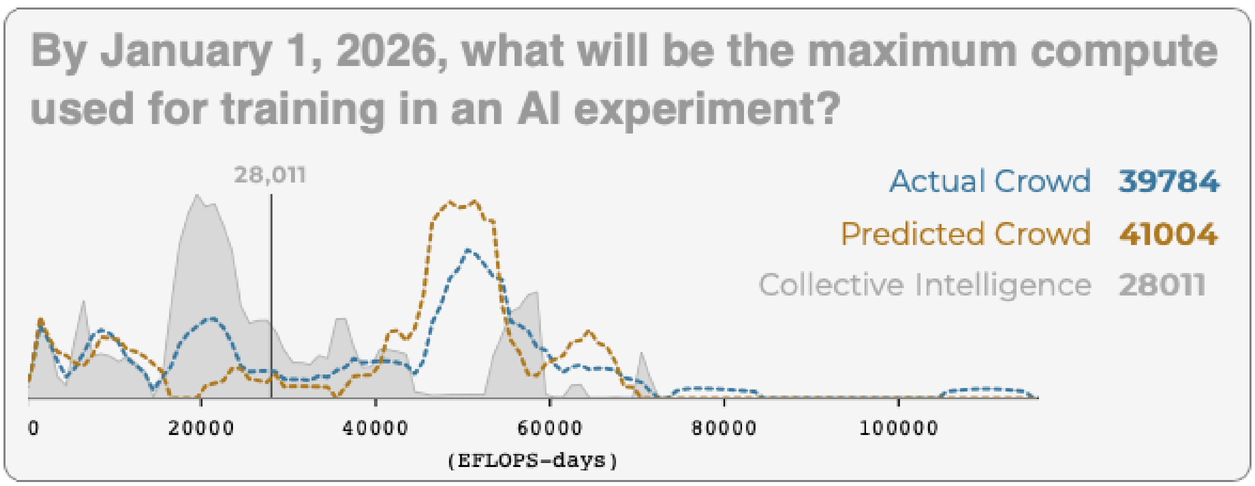

FIGURE 2 *- An example of prediction-normalized forecast (i.e.,"collective intelligence" in grey) extracted from the aggregation of 59 personal forecasts ("actual crowd" in blue) and 59 meta forecasts ("predicted crowd" in orange) for a continuous target variable measured in exaFLOPS-days. Other examples taken from Hypermind's forecasting contest on Artificial Intelligence 2030 are showcased here.*

Practical approximations

In practice, when applying this algorithm to experimental data obtained with a relatively small number of forecasters - for instance just 59 forecasters in the case presented in Figure 2 - several approximations are necessary to produce coherent results.

A first design decision concerns the granularity of the bins along the target variable's range. This immediately impacts the number of bins, which in turn impacts the amount of probability that is allotted in each bin by the forecasters. It's important to avoid very small probabilities that can potentially transform the *meta(j|i)/meta(i|j)* ratios in equation (2) into huge numbers. We decided to set the bin granularity to the minimum standard deviation allowed for input distributions (e.g., those in figure 1), which is specific to a target variable range. This would typically result in a few hundred bins, which is more than a forecaster could reasonably distinguish amongst. This seems to offer some degree of psychological coherence.

A second approximation is the minimum probability required to consider that a forecaster has a "non-zero" probability in a particular bin. This goes to the core of equation (2) which computes the prediction-normalized forecast based on which participants have a forecast or a meta forecast in each bin. After some experimentation, we settled on a rather aggressive threshold of .05 probability below which a forecaster is not considered to have forecasted or meta-forecasted that particular bin.

A third and perhaps less critical approximation is also related to the issue of avoiding very small meta probabilities that can potentially transform the *meta(j|i)/meta(i|j)* ratios in equation (2) into huge numbers. This in turn tends to extremize the prediction-normalized forecast density into very high peaks and very low valleys, which can result in apparent "holes" in the density function where probabilities are almost zeroed-out. Avoiding those "holes"simply requires that the *meta(i|j)* denominator be no less than a minimum value, typically .001. The values of this parameter have very little impact on the surprisingly-popular point forecast, but relatively high values such as .001 (vs .0001 for instance) do help fill up the density function properly.