Results

Abhinav Tushar

April 13, 2018

This document contains some notes on the results. This is not a definitive text covering everything and might contain only the minimal results required to provide an explanation for a few things we are concerned about. To reproduce all of the results, use the notebooks.

Contents

1 Experiments

The experiments in this repository are aimed to answer the following general questions in context of modeling a weight based ensemble for live prediction of influenza:

1. How far can we push a weight based ensemble? What is the upper limit given a certain set of component models?

2. Can we break down the time points in a season in identifiable chunks where we can then learn different set of component weights? If yes, how much improvement do we get over a single set of weights for the whole season?

3. Since a single model which has been really good in the past can be overtaken in another season (depending on how different is this season or how much the other models have improved), should we be using the same set of weights for the whole season?

Next subsections elaborate on the approaches to get an answer to these questions:

1.1 Upper limit on score

In all the experiments, the prediction of our ensemble at any point in time is just a weighted sum of corresponding predictions of the components. Since a *prediction* is just a probability distribution, a direct way to get an upper bound on the possible score is to just check the score of the best component at each time point.

Intuitively, an oracle ensemble would know which component model is the best at each time point and would assign a weight of 1 to that while 0 to all the others. The score of such an oracular model is what we present as the upper limit in this document.

1.2 Optimal segments

For the second question, we exhaustively look at all possible segments in the time of a season (around 33 time points, each mapping to a week in season). For each segment from all the training seasons, we learn weights for all the components.

Notice that when $k = 1$, we have a model that just learns one set of weights for the whole season. When $k = 33$, we have a model that learns one weight for each week. This $k = 33$ model will have training scores very close[1] to the oracular model since it can learn weights like $[1, 0, 0, \ldots]$.

If a $k > 1$ model here does well on cross validation, we can answer the second question by saying that there are these segments in which model performances vary sufficiently or something along those lines.

1.3 Dynamic weights

Even if there are no established segments in a season, we might have unseen situations where a assigning higher weights to a certain model which has not been that good in the past might payoff.

[1]Though not exactly since we are learning over a bunch of seasons while the oracular model goes separately on each season

Note that we are not putting any assumption on the component models themselves. A component model is totally free to evolve and its performance can go up or down depending on its own whim. We may as well assume it to be controlled by an adversary[2] which might fool us by doing good in training while doing really bad when we start predicting. In any case, if we have a scheme to change the weights dynamically, we will have (in a way) a more robust ensemble.

Our dynamic weight ensemble should have low regret (the difference between its performance and the best single component in hindsight). In our experiments, we use hedging ([1]) which provides good upper bound on regret.

2 Some findings

- Multiplicative weights does better than simple mean when started with equal weights. If started with weights from degenerate em, it does better than dem (3/4) on test data.

Target	Upper limit	Mean	DEM	k-DEM	MP
1-ahead	-1.8314	-2.7822	-2.6189	-2.6189 (k=1)	**-2.6182**
2-ahead	-1.8883	-2.9742	-2.8250	**-2.8225** (k=7)	-2.8241
3-ahead	-2.0386	-3.1339	**-2.9931**	**-2.9931** (k=1)	-2.9948
4-ahead	-2.1614	-3.2481	-3.0996	-3.09961 (k=1)	**-3.0986**

- Doesn't look like there are well defined regions (in the time axis) where specific models outperform some other consistently. *Deduced from the results using k partition dem models.*

- The dimension where we possibly can quantify the role of different models might actually be the y axis (wili). There might be more well defined discretizations here than on the time axis since the differences between statistical and mechanical models come into play here. For example in the high (and anomalous) time of this season, at one week we didn't have delphi and reich lab models (which are mostly statistical), the partial ensemble predicted better because of the CU's mechanistic models.

- Since dynamic weight models implicitly take into account this y axis variability, they should be more robust for our purpose.

[2]This also captures the worst case uncertainty in the time series itself.

- This season's results

Target	Upper limit	Mean	DEM	k-DEM	MP
1-ahead	-2.2603	-3.3267	-3.2496	-3.2496 (k=1)	**-3.2461**
2-ahead	-2.6580	-3.8180	-3.8424	-4.0286 (k=7)	-3.8546 (**-3.8060** with reset)
3-ahead	-2.9698	-4.1643	-4.2478	-4.2478 (k=1)	-4.2648 (**-4.1628** with reset)
4-ahead	NA	NA	NA	NA	NA

When started with weights from DEM model, MP doesn't do good. If instead started with equal weights (*with reset*), models with less weights get a chance to express and give results in anomalous season (where mean is better than dem).

3 What now?

So there are no specific identifiable segments but we do get something when we use dynamic weights. Specifically, if we put the starting weights right for the MP model, we can get stabler performance. To do this we can use a fixed share scheme [2] which adds another parameter α and avoids assigning near zero weights to models. Or simply put an interpolating parameter between equal weights and dem weights.

References

[1] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[2] Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.