Hedging bounds

April 6, 2018

Contents

1 Introduction

Hedging is a multiplicative weight algorithm for dynamically updating resource allocation among experts resulting in low regret of the weighted model with respect to the best performing model in the mix. The losses here are calculated over a set of true values released sequentially at each time step t upto a final time of T. In the general setting, at each time step, we get a *final* truth which helps in finding the loss of the each expert and the ensemble as a whole. In this document, we try to analyze hedging in a setting where the true values revealed at each time are not exact and are updated by a set of patches released at the next few time steps. In this setting, there are two losses for a model at each time step. One, $\hat{m}$, tells us how the model does with respect to the first estimate of truth and second, m, is the model loss with respect to the *final truth.* Our aim here is to bound the final truth loss of the ensemble in terms of the final truth loss of the components and try to find ways to improve on that.

2 TODO Problem setting

3 Bounds

Suppose the true time series is $s(t)$. Due to data revisions, we first get a rough estimate $\hat{s}(t)$ and then a set of patches $\delta_j(t)$ at j time points after t where j goes from 1 to some k. Its mostly safe to assume that these patches are normally distributed with $|\mu_j| > |\mu_{j-1}|$ (though this information is not used anywhere as of now).

Because of these updates in s, we get certain uncertainty in estimating the model loss $m_i(t)$. Let $\hat{m}_i(t)$ be the first estimate of loss, then $m_i(t)$ (the real loss) is bounded as:

$$\hat{m}_i(t) - \sum_{j=1}^{k} \theta_i(j) \leq m_i(t) \leq \hat{m}_i(t) + \sum_{j=1}^{k} \theta_i(j)$$

Here $\theta_i(j)$ captures the effect of δ_j for the i^{th} model.

An online ensemble strategy here describes the weight update mechanism based on losses that each of the models and the ensemble as a whole receives. What we want is to bound the ensemble loss $L_H = \sum_{t=1}^{T} \sum_{i=1}^{n} p_i(t) m_i(t)$ in terms of the loss of any single expert $L_i = \sum_{t=1}^{T} m_i(t)$. Here $p_i(t)$ is the normalized weight $w_i(t)$ for expert i at time t. Summing over the relation between $\hat{m}_i(t)$ and $m_i(t)$, we get the following relations between L and $\hat{L}$ ($\hat{L}$ is the sum of loss based only on the first estimate).

$$\hat{L}_i - T \sum_{j=1}^{k} \theta_i(j) \leq L_i \leq \hat{L}_i + T \sum_{j=1}^{k} \theta_i(j)$$

$$\hat{L}_H - T \max_{i' \in [1 \ldots n]} \sum_{j=1}^{k} \theta_{i'}(j) \leq L_i \leq \hat{L}_i + T \max_{i' \in [1 \ldots n]} \sum_{j=1}^{k} \theta_{i'}(j)$$

3.1 Regular hedging bound

If we do regular hedging and have the real time series in hand, then the bound is similar to [1] and is given by:

$$L_H \leq \frac{-\ln w_i(1) - L_i \ln \beta}{1 - \beta}$$

β is a hyper parameter in $[0, 1]$.

3.2 Hedging using only the current data

Since we don't have real data, we can only use the estimates. The weight update equation is $w_i(t+1) = w_i(t)\beta^{\hat{m}_i(t)}$. If we only use the latest estimate without utilizing the patches we get for earlier time points, we get the following bound which adds the extra uncertainty term:

$$L_H \leq \frac{-\ln w_i(1) - L_i \ln \beta}{1-\beta} - \frac{\ln \beta}{1-\beta} T(\sum_{j=1}^{k} \theta_i(j) + \max_{i' \in [1 \ldots n]} \sum_{j=1}^{k} \theta_{i'}(j))$$

The bad thing here is that the extra term is dependent on T which makes it poorer as time increases.

3.3 Hedging by live estimate update

Here, we use all data available to us at any moment. This is equivalent to recalculating the weights from the start (using $w_i(1)$ values) at every time step. The weight updates here follow the following inequalities:

$$\begin{aligned} w_i(T+1) &\geq w_i(1)\beta^{\sum_{t=1}^{T-k} m_i(t)} \beta^{\sum_{t=T-k+1}^{T} \hat{m}_i(t)} \beta^{\sum_{j=1}^{k-1} \theta_i(j)(k-j)} \\ w_i(T+1) &\leq w_i(T)\beta^{m_i(T)} \beta^{-\sum_{j=1}^{k} \theta_i(j)j/T} \end{aligned}$$

This time we get the following bound:

$$L_H \leq \frac{-\ln w_i(1) - L_i \ln \beta}{1-\beta} - \frac{\ln \beta}{1-\beta} \left(\left(\sum_{j=1}^{k} \theta_i(j)(2k-j) \right) + \max_{i' \in [1 \ldots n]} \left(\sum_{j=1}^{k} \theta_{i'}(j)j \right) \right)$$

This doesn't involve T and thus is asymptotically better. Another thing to note here is that the term involving max says that we can do better by removing models with high θ values.

There are a few things to note regarding the component models:

- To reduce the loss L_H, one can put a good model in the mix (with low L_i) but since the models actually experience real time loss of $\hat{L}_i$, they will mostly have some θ values to trade off.

- A very accurate oracle model will have almost zero L_i but to do that, it will have to lower its $\hat{L}_i$ and thus will have higher θ values.

- A flatter model (like a uniform probability one) will have $\theta_i(j) = 0$ but will have high L_i.
- The model which is bad with lags (resulting in the max term) and is not the best one considering its L_i can be removed without any theoretical loss in performance.

3.4 Hedging by preempting the estimate

To reduce the effect of truth revisions, we can update weights *not* based on the first performance estimate but on an offset value which depends on the history of the lags. This basically means to add an estimate for the error $\hat{m} - m$ in the current $\hat{m}$.

The practical problem here is that we probably don't have enough time points. Anyway, effectively what we are after is to minimize the total error we accumulate by this estimation, which is $\left(\sum_{t=1}^{T} x(t)\right) - E[\hat{m}(t) - m(t)]$. Here $x(t)$ is our estimate at time t. If $\hat{m}(t) - m(t)$ follows a distribution then a nice way to estimate this is to just use the mean of whatever truth values we know about:

$$x(t) = \frac{1}{t'} \sum_{i=1}^{i<t'} \hat{m}(i) - m(i)$$

Here, t' is the time $< t$ for which we have full data.

However, if the distribution is shifting, a better way might be to update estimate using an α mixing parameter like shown below:

$$x(t+1) = \alpha x(t) + (1 - \alpha)(\hat{m}(t) - m(t))$$

If there are systematic distribution tendencies in the model (see next section) it is going to be reasonable, empirically, to initialize the error offsets using the mean values of past points and go preempting from there on.

4 Understanding $\hat{m} - m$

In the current formulation, θ sets the upper limit on how much a model's final loss can differ from its first estimate loss. Since each model provides a distribution as its prediction, all θ values are naturally upper bounded by 1.

$$\sum_{j=1}^{k} \theta_i(j) = \max_{t \in [1..T]} |m_i(t) - \hat{m}_i(t)|$$

Each model's lag performance can be specified by its loss values in the shaded region of figure 1. Such plots might help us figure out models to keep in the tracking ensemble and also help in understanding the θ values.

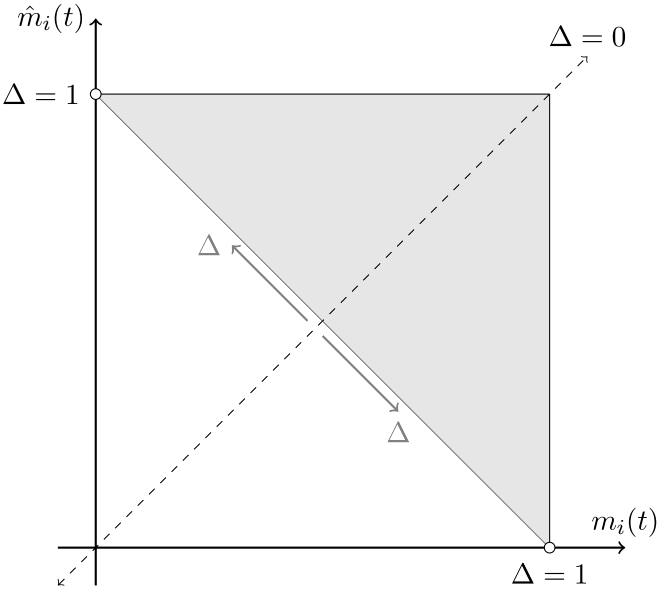

Figure 1: $m - \hat{m}$ plot. A model really good with the actual truth will have m_i values close to 0 but will have high Δ ($= |\hat{m}_i - m_i|$). A model which does really good on first estimate of truth will have low $\hat{m}_i$ but will also have high Δ. All lines parallel to $\hat{m}_i = m_i$ denote a single value of Δ. Δ is 0 at $\hat{m}_i = m_i$ and grows on both sides as shown. *Note that only the shaded portion is the valid region for points to lie in since we can constraint m values using the fact that they are generated from the complement of a probability distribution. However, the points can also lie outside the region if they are on the $x = y$ line, meaning the observed and actual truth were the same.*

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.