Super Learning in Joint Models
First brief report on the references

Arnau García

September 9, 2024

This brief report is intended to be a summary of all the references I have been consulting. In this summary I will focus mainly in the Super Learner concept [11], trying to adapt the notation and the problem to the Joint Model paradigm [8].

To familiarize myself with the concept of Super Learning from a theoretical point of view I have been reading [11], [7] and [1]. Moreover, to explore super learners from a more practical point of view I have studied [6], as well as [5]. Finally, to understand Super Learning in the Joint Modeling context I have consulted [10].

From now on, we will be using the notation in [10]. Assume we have the data $\mathcal{D}_n = \{T_i, \delta_i, \boldsymbol{y}_i; i = 1, \dots, n\}$. Where T_i^* denotes true time to true event of interest ε for the i-th subject, C_i the censoring time, $T_i = min(T_i^*, C_i)$ is the corresponding observed time to the event, and $\delta_i = \mathbf{1}(T_i^* \leq C_i)$ is the event indicator. Moreover, $\boldsymbol{y}_i$ is the $n_i \times 1$ longitudinal response vector for the i-th subject, with element y_{il} being the value of the longitudinal outcome taken at time point t_{il}, $l = 1, \dots, n_i$ (n_i is the number of measurements for the i-th subject). We work under the general framework postulated in [10], we assume that the response vector $\boldsymbol{y}_i$ conditional on the random effects $\boldsymbol{b}_i$ has distribution $\mathcal{F}_\Psi$ parameterized by the vector Ψ. By taking this formulation we allow for distributions not covered by the exponential family.

We will be working under the Bayesian approach, inferences are drawn via the joint posterior distribution $\{\boldsymbol{\theta}, \boldsymbol{b} | \boldsymbol{Y}, \boldsymbol{T}, \boldsymbol{\delta}\}$, where $\boldsymbol{\theta}$ is the vector of all model parameters. Find more details on the parameters of the model, as well as the general model specification in [10].

Assume we have the library $\mathcal{L} = \{M_1, \dots, M_L\}$ composed by L different joint models. When working with several Bayesian models and the interest relies on improving predictions, a widely used method is the Bayesian Model Averaging (BMA) [1]. Nonetheless, BMA has some

[1] I have done some work related with BMA, you can see it in `https://github.com/ArnauGF/MSc-UPC/tree/bayesian`

drawbacks. First, it is necessary to compute the marginal likelihood under each model:

$$p(\mathcal{D}_n|M_l) = \int p(\mathcal{D}_n|\boldsymbol{\theta}_l)p(\boldsymbol{\theta}_l|M_l)d\boldsymbol{\theta}_l. \tag{1}$$

The previous computation is not that easy, in general it is computationally expensive, see a discussion on how to compute (1) in [2]. Moreover, the BMA weights are not explicitly designed to optimize the accuracy of predictions, and it is unclear if the method accounts in a proper manner for over-fitting. The concept of *super learning* (SL) is the chosen one to overcome the issues presented above. The super learner is a prediction method designed to find the optimal combination of a collection of prediction algorithms. Optimality is defined with respect an objective function (for instance, minimizing the mean squared error, maximizing the area under the receiver operating characteristic curve (ROC), etc) specified by the user. The super learner framework is built on the theory of cross-validation and allows for a general class of prediction algorithms to be considered for the ensemble.

So, let us expose the super learning procedure in the Joint Modeling context. Since we are working under the cross-validation theory, the first step is to split the original dataset $\mathcal{D}_n$ into V subsets. The choice of V depends on different items, such as the number of events in the dataset. Due to the fact that our interest relies on predictions with a dynamic nature (in longitudinal profiles, as well as in time to event data, it is clear that time plays an important role) we want to derive the best combination of predictors in different follow-up times. Thus, we consider the sequence of time points $t_1, \ldots, t_Q$. The number and placing of these times will depend again on several items. For any $t_q \in \{t_1, \ldots, t_Q\}$ we define $R(t_q, v)$ to be the set of subjects at risk at time t_q belonging to the vth fold. For all the subjects in $R(t_q, v)$, we calculate the cross-validated predictions. Using the cross-validation method, we fit the L models in the combined $v-1$ folds, and we will calculate predictions for the vth fold that were left out. In the following expressions we are conditioning on the covariates $\boldsymbol{w}_i, \boldsymbol{x}_i$ and $\boldsymbol{z}_i$, but it is not explicitly denoted to simplify notation,

$$\hat{\pi}_i^{(v)}(t_q + \Delta t|t_q, M_l) = P\left(T_i^* < t_q + \Delta t|T_i^* > t_q, \mathcal{H}_i(t), M_l, \mathcal{D}_n^{(-v)}\right). \tag{2}$$

The above calculation is based on the joint model M_l in $\mathcal{L}$, and this model was fitted in the data set $\mathcal{D}_n^{(-v)}$ that excludes the subjects in the vth fold. Notice that the calculation is based on a Monte Carlo approach, where the posterior distributions of the parameters under M_l are used. In addition, the sub-index i is denoting that that cross-validated prediction in (2) is for the ith subject. Let us assume that we have r_v subjects at risk at time t_q within the vth fold, and we write $R(t_q, v) = \{1_v, \ldots, r_v\}$ to denote these subjects. Assume we have fixed the timepoint t_q, then, for the different models and the different subsets of data we can create the following matrix to to gather all the predictions we have:

$$\Pi(t_q) = \begin{pmatrix} \hat{\pi}^{(1)}_{1_1}(t_q + \Delta t|t_q, M_1) & \cdots & \hat{\pi}^{(1)}_{1_1}(t_q + \Delta t|t_q, M_L) \\ \vdots & \ddots & \vdots \\ \hat{\pi}^{(1)}_{r_1}(t_q + \Delta t|t_q, M_1) & \cdots & \hat{\pi}^{(1)}_{r_1}(t_q + \Delta t|t_q, M_L) \\ \vdots & \ddots & \vdots \\ \hat{\pi}^{(V)}_{1_V}(t_q + \Delta t|t_q, M_1) & \cdots & \hat{\pi}^{(V)}_{1_V}(t_q + \Delta t|t_q, M_L) \\ \vdots & \ddots & \vdots \\ \hat{\pi}^{(V)}_{r_V}(t_q + \Delta t|t_q, M_1) & \cdots & \hat{\pi}^{(V)}_{r_V}(t_q + \Delta t|t_q, M_L \end{pmatrix}, \tag{3}$$

where we have $\#R(t_q) = \sum_{v=1}^{V} r_v$ (the number of individuals at risk at time t_q in the whole dataset) rows and L columns. Normally, when using the super learner scheme, we end up with a similar matrix than the one above, but with a dimension of $n \times L$ (i.e. the number of rows is the number of observations in the whole dataset), see [6]. Nonetheless, this is not the case in our context. In the survival framework we are not interested in compute predictions for those individuals out of the risk set. Of course, does not make sense to predict the conditional risk of suffering ε at time t_q for those individuals who have already suffered the event. Now, summing by columns the matrix above we define the following quantity

$$\hat{\tilde{\pi}}^{v}_i(t_q + \Delta t|t_q) = \sum_{l=1}^{L} \tilde{w}_l(t_q)\hat{\pi}^{(v)}_i(t_q + \Delta t|t_q, M_l), \quad \text{for all } v \in 1, \ldots, V, \tag{4}$$

with the constraints $\tilde{w}_l(t_q) > 0$ for $l = 1, \ldots, L$ and $\sum_l \tilde{w}_l(t_q) = 1$. And then, due to the constraints we have chosen, (4) is the convex combination of the L predictors in each row of $\Pi(t_q)$. Of course, the weights $\tilde{w}_l(\cdot)$ are time-dependent. The weighted combination of the L predictors is called the *ensemble super learner* (eSL), while the model with the best cross-validated prediction metric is the so-called *discrete super learner* (dSL).

As said before, weights $\{\tilde{w}(t_q), l = 1, \ldots, L\}$ are chosen such that optimality is achieved. Now, it is time to define optimality in this context. Following [10], we work under the *proper scoring* framework. I have used [4] and [3] to familiarize myself with proper scoring. Later, we will add some comments on proper scoring and a couple of scoring rules that can be used here. A scoring rule $\mathcal{S}\{\pi_i(u|t), \mathbf{1}(t < T^*_i < u)\}$ is said to be proper if

$$E\left[\mathcal{S}\{\pi^{\text{true}}_i(u|t), \mathbf{1}(t < T^*_i < u)\}\right] \leq E\left[\mathcal{S}\{\hat{\pi}_i(u|t), \mathbf{1}(t < T^*_i < u)\}\right], \quad u > t,$$

is satisfied for every estimate $\hat{\pi}_i(u|t)$ of $\pi^{\text{true}}_i(u|t)$, the latter denoting the conditional risk probabilities under the true model. Notice that the scoring rule above, $S(\cdot, \cdot)$, is defined such

that the lower the score the better the accuracy. Thus, weights $\{\tilde{w}(t_q), l = 1, \ldots, L\}$ are the ones minimizing the selected proper scoring rule of the cross-validated predictions

$$\hat{\tilde{w}}(t_q) = \arg\min_{\tilde{w}(t_q)} \left[\sum_{l=1}^{V} \mathcal{S}\{\hat{\tilde{\pi}}_i^v(t_q + \Delta t|t_q), T_i, \delta_i\} \right], \tag{5}$$

under the constraints exposed before ($\tilde{w}_l(t_q) > 0$ for $l = 1, \ldots, L$ and $\sum_l \tilde{w}_l(t_q) = 1$) and where $\mathcal{S}(\cdot, \cdot)$ is a proper scoring rule. Once we have the weights, computed with respect the desired criterion, then we can easily build the eSL by taking the weighted sum (with the corresponding weights) of the predictions, this time fitted with the whole data available.

Having exposed the super learning procedure in the joint models context, now we want to present several properties that are relevant for our problem. First of all, we want to comment why the constraints $\tilde{w}_l(t_q) > 0$ for $l = 1, \ldots, L$ and $\sum_l \tilde{w}_l(t_q) = 1$ for the weights are taken. As Breiman discuss in [1], taking into account that the predictions derived from the different algorithms may be strongly correlated (we expect similar predictions if the algorithms are working), by using a general linear combination $\sum_l \tilde{w}_l(t_q)\hat{\pi}_i(t_q + \Delta t|t_q, M_l)$ to get a new predictor there is no guarantee that the resulting predictor will stay near the range $[\min_l \hat{\pi}_i(t_q + \Delta t|t_q, M_l), \max_l \hat{\pi}_i(t_q + \Delta t|t_q, M_l)]$, and then the generalization of the method may be poor. Nonetheless, by taking the constraints above, we have that

$$\min_l \hat{\pi}_i(t_q + \Delta t|t_q, M_l) \leq \sum_l \tilde{w}_l(t_q)\hat{\pi}_i(t_q + \Delta t|t_q, M_l) \leq \max_l \hat{\pi}_i(t_q + \Delta t|t_q, M_l), \tag{6}$$

and so, the convex combination of the predictors is more stable, less sensible to small changes in the data.

Another important consideration, commented in [10] as well as [6], is that when the eSL is used, it is recommended to evaluate it as another candidate in a dSL. If the eSL performs better than any other candidate, the dSL will end up selecting the eSL.

Now, let us comment on the so-called *oracle inequalities*, exposed in [11] as well as in the set of slides [2].We define the oracle estimator as the best possible estimator given the set of candidates considered, however, it depends on both the observed data and the truth distribution generating the data, and thus the oracle estimator is unknown. The oracle inequalities state that given a couple of assumptions (having an uniformly bounded loss function plus another bounding hypothesis, see more details in [11] and the set of slides), the super learner performs as well as the oracle selector. In addition, as long as the number of candidate learners considered is polynomial in sample size, the super learner is the optimal learner.

[2]See the set of slides made by E. Polley in `https://www.stat.berkeley.edu/users/laan/Class/Class_subpages/BASS_sec1_3.1.pdf`

The oracle inequalities have been extended to allow the candidate learner selector to include weighted averages of the candidate learners, and the obtained results are similar. Then, one can conclude that the super learner has good properties, in the sense it is performing as the best algorithm. Moreover, this is proved by means of a cross-validation theorem and by taking some assumptions on the loss function.

Finally, we would like to present a brief discussion about the scoring rules. There are several possibilities when looking for proper scoring rules, some of they are exposed in [3] and [4]. Nonetheless, the ones exposed in [10] are the Brier score, as well as the integrated Brier score,

$$\begin{aligned} BS(t+\Delta t, t) &= E[\{\mathbf{1}(T_i^* \leq t+\Delta t) - \hat{\tilde{\pi}}_i(t+\Delta t|t)\}^2 | T_i^* > t] \\ IBS(t+\Delta t, t) &= \frac{1}{\Delta t}\int_t^{t+\Delta t} BS(s,t)ds, \end{aligned} \tag{7}$$

the Brier score evaluates the predictive accuracy at time $t+\Delta t$, and the integrated Brier score summarize the predictive accuracy in the interval $(t, t+\Delta t)$. Moreover, an adaptation of the expected predictive cross-entropy (EPCE) is presented as an alternative proper scoring rule in [10]. While the Brier score is easier to interpret than the EPCE, the latter present the advantage that we can easily compute an estimate that account for censoring.

Questions

I would like to add a final section adding some questions I have.

- Having studied about super learning, as well as about joint modeling, of course, I have some doubts about where I have to study the adaptation of super learning. There are two types of dynamic individualized predictions, as it is exposed in [9]. These predictions are the one associated to the probability of survive up to a time, say u, and the one associated to predict the longitudinal response of a given patient

$$\begin{aligned} \pi_j(u|t) &= P(T_j^* \geq u | T_j^* > t, \mathcal{Y}_j(t), \boldsymbol{w}_j, \mathcal{D}_n) \\ \omega_j(u|t) &= E[y_j(u) | T_j^* > t, \mathcal{Y}_j(t), \mathcal{D}_n]. \end{aligned} \tag{8}$$

  However, in [10], we are working with the cross-validates predictions (2), which are quantities that seem to be more linked with the instantaneous risk of event, ie the hazard function. If working with (8) I assume that some changes would be added, at least scoring rules as presented should be adapted. Nonetheless, I see in `https://drizopoulos.github.io/JMbayes2/articles/Super_Learning.html` that weights are computed with respect (2) and then are applied to compute the predictions (8). So, I am not fully understanding this, how the same weights optimize the predictions for longitudinal outcomes and survival probabilities?

References

[1] Leo Breiman. Stacked regressions. *Machine learning*, 24:49–64, 1996.

[2] Siddhartha Chib. Marginal likelihood from the gibbs output. *Journal of the american statistical association*, 90(432):1313–1321, 1995.

[3] Alexander Philip Dawid and Monica Musio. Theory and applications of proper scoring rules. *Metron*, 72(2):169–183, 2014.

[4] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[5] Ashley I Naimi and Laura B Balzer. Stacked generalization: an introduction to super learning. *European journal of epidemiology*, 33:459–464, 2018.

[6] Rachael V Phillips, Mark J Van Der Laan, Hana Lee, and Susan Gruber. Practical considerations for specifying a super learner. *International Journal of Epidemiology*, 52(4):1276–1285, 2023.

[7] Eric C Polley and Mark J Van der Laan. Super learner in prediction. *Statistical applications in genetics and molecular biology*, 2010.

[8] Dimitris Rizopoulos. *Joint models for longitudinal and time-to-event data: With applications in R*. CRC press, 2012.

[9] Dimitris Rizopoulos, Laura A Hatfield, Bradley P Carlin, and Johanna JM Takkenberg. Combining dynamic predictions from joint models for longitudinal and time-to-event data using bayesian model averaging. *Journal of the American Statistical Association*, 109(508):1385–1397, 2014.

[10] Dimitris Rizopoulos and Jeremy MG Taylor. Optimizing dynamic predictions from joint models using super learning. *Statistics in Medicine*, 43(7):1315–1328, 2024.

[11] Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007.