Simulation protocol

Judith Neve

December 11, 2022

1 Studies

1.1 Aims

1.1.1 Study 1: Hyperparameters to tune

Prior findings [1] have shown the number of predictors considered at a split and the sample fraction to be the two most influential hyperparameters on model accuracy. However, these findings only investigate the effect of tuning one or two hyperparameters at once. This study aims to extend these findings by considering more combinations of hyperparameters in order to identify the combination of hyperparameters for which tuning leads to the best predictive performance of a prediction model.

1.1.2 Study 2: Optimisation metric

This study aims to identify the metric to optimise in the tuning procedure which leads to the best predictive performance of a prediction model. We tune the combination of hyperparameters considered to be the most optimal in Study 1.

1.1.3 Study 3: Hyperparameter search algorithm

This study aims to identify the hyperparameter search algorithm which leads to the best model performance. We tune the combination of hyperparameters considered to be the most optimal in Study 1 and optimise the metric considered optimal in Study 2.

1.2 Data-generating mechanism

1.2.1 Population

Different datasets will be generated for each of the three studies. A full factorial simulation design will be used to consider the influence of data characteristics on tuning procedures. The varying factors will be the number of candidate predictors p, the event fraction EF, and the sample size N. The levels of these three factors are detailed in Table 1. A total of 27 (3*3*3) scenarios will be

Table 1: Data generating scenarios.

Characteristics	Levels
Number of candidate predictors	8, 16, 32
Event fraction	0.1, 0.3, 0.5
Sample size	$0.5n$, n, $2n$

n refers to the minimum sample size required to identify effects for a given number of regression coefficients (here, $1.25p$) and expected event fraction [2] with an AUC of 0.8. This is obtained using the R package `pmsampsize`.

considered. 1,000 datasets will be generated for each scenario, yielding a total of 27,000 datasets per study.

Development and validation data will be simulated under a logistic model with strong interactions. For each observation i (i = 1, ..., N), predictors $\mathbf{x}_i$ will be drawn from a p-variate normal distribution with parameters detailed in Formula 1.

$$\mathbf{x}_i \sim \text{MVN}(\mathbf{0}, \begin{bmatrix} 1 & 0.2 & ... \\ 0.2 & 1 & ... \\ ... & ... & ... \end{bmatrix}) \tag{1}$$

Additionally, $0.25p$ two-way interactions will be computed, with the j^{th} interaction being the product of the j^{th} and the $(j + p/2)^{th}$ predictors. Then, the binary outcome y_i will be drawn from a Bernoulli distribution conditional on $\mathbf{x}_i$, computed interactions, and the regression coefficients for main and interaction effects of the data generating model, hereafter called "true effect" (Formula 2).

$$P(y_i = 1) = \frac{exp(\beta_0 + \beta * \sum_{j=1}^{p} x_{ij} + \gamma * \sum_{j=1}^{0.25*p} x_{ij} * x_{i(j+0.5p)})}{1 + exp(\beta_0 + \beta * \sum_{j=1}^{p} x_{ij} + \gamma * \sum_{j=1}^{0.25*p} x_{ij} * x_{i(j+0.5p)})} \tag{2}$$

A validation dataset (N = 10,000) will be generated for each event fraction and number of candidate predictors combination in order to evaluate model performances. In the most extreme scenario ($EF = 0.1, p = 32$), this yields $\frac{10{,}000*0.1}{1.25p} = 25$ events per variable, which is well above the 10:1 events per variable rule of thumb.

1.2.2 True effect estimation

True effects will be constant across studies. They will be determined as follows: for each combination k of number of candidate predictor and event fraction, the intercept $\beta_0^{(k)}$, predictor main effects $\boldsymbol{\beta}^{(k)}$, and predictor interaction effects $\boldsymbol{\gamma}^{(k)}$ will be estimated using a large sample (N = 100,000) approximation. All main effects ($\boldsymbol{\beta}^{(k)}$) and interaction effects ($\boldsymbol{\gamma}^{(k)}$) will be set to be equal (i.e., $\beta_1^{(k)} = \beta_2^{(k)} = ... = \beta_p^{(k)}$ and $\gamma_1^{(k)} = \gamma_2^{(k)} = ... = \gamma_{0.25p}^{(k)}$). The estimation will use the R function `optim`, focused on minimising a loss function measuring the sum of i) the absolute difference between the targeted AUC and the observed AUC in the simulated dataset, and ii) the absolute difference between the targeted event

fraction and the average estimated probability $P(y_i = 1|\mathbf{x}_i)$ in the simulated dataset. This estimation will be done in three steps:

1. Optimise $\beta_0^{(k)}$ and $\boldsymbol{\beta}^{(k)}$ for a target AUC of 0.7.

2. Using the optimised $\beta_0^{(k)}$ and $\boldsymbol{\beta}^{(k)}$ from step 1, optimise $\boldsymbol{\gamma}^{(k)}$ for a target AUC of 0.8, such that a model ignoring interactions would have an AUC of 0.7 while including the correct interactions would lead to an AUC of 0.8. $\boldsymbol{\gamma}^{(k)}$ will be constrained to be positive.

3. Using the optimised $\boldsymbol{\beta}^{(k)}$ and $\boldsymbol{\gamma}^{(k)}$, optimise $\beta_0^{(k)}$ for a target AUC of 0.8 to ensure the interactions do not alter the event fraction.

This will be repeated 20 times and the mean of the parameters will be taken to obtain more stable estimates.

Results from this numerical procedure for $\beta_0^{(k)}$, $\boldsymbol{\beta}^{(k)}$ and $\boldsymbol{\gamma}^{(k)}$ will be checked using an independently generated dataset of N = 1,000,000. It will be checked whether:

1. The observed event fraction is at a distance of at most 0.01 from the target event fraction.

2. The AUC of a model ignoring the interaction terms is at a distance of at most 0.025 from 0.7.

3. The AUC of a model including the interaction terms is at a distance of at most 0.05 from 0.8.

4. The estimated coefficients when fitting a logistic regression model are at a distance of at most 0.05 from the coefficients used to generate the dataset.

1.3 Estimands

All studies focus on predictive performance for dichotomous outcome models. We also evaluate the computational time for each tuning procedure.

1.4 Methods

1.4.1 Study 1: Hyperparameters to tune

We will vary which hyperparameters are tuned when fitting a random forest using the R package `ranger` via the R package `caret`. We will use grid search (as is the standard in this package) to optimise classification accuracy at a probability threshold of 0.5 (as is the default in `caret`). 5-fold cross-validation will be used as part of the tuning procedure.

[1] found `mtry` (the number of predictors randomly sampled to make a split) and `sample.fraction` (the proportion of the data that is used to fit a single tree) to be the pair of predictors with the highest influence on accuracy. We would therefore suggest to always tune these two hyperparameters. However, [3]

Table 2: Hyperparameter tuning ranges

Hyperparameter	Default	Range
`mtry`	$\sqrt{p}$ (rounded down)	1-p
`min.node.size`	1	1-10
`num.trees`	500	100, 200, ..., 1000
`replace`	TRUE	TRUE, FALSE
`sample.fraction`	1	0.1, 0.2, ..., 0.9, 1
`splitrule`	gini	gini, hellinger, extratrees

For `splitrule` = extratrees, an additional parameter should be considered regarding the number of random splits to consider. This will be set to its default of 1.

demonstrates that `sample.fraction` has a similar effect to and `min.node.size` (that is, the minimum number of observations for a node to be formed, i.e., the point at which a split should not be computed regardless of impurity). As `caret` allows tuning for `min.node.size` but not for `sample.fraction` in its settings, we opt to always tune `mtry` and `min.node.size` to increase to user-friendliness of our possible findings.

All combinations of the following hyperparameters will be tuned in conjunction with `mtry` and `min.node.size`:

- `num.trees`, that is, the number of trees the random forest fits and therefore averages over.
- `replace`, that is, whether the data used to fit a single tree is sampled with or without replacement.
- `sample.fraction`, that is, the proportion of the data that is used to fit a single tree.
- `splitrule`, that is, the way in which a split is picked.

Default values and tuning ranges are presented in Table 2.

This leads to 16 ($\sum_{h=0}^{4} \binom{4}{h}$) different combinations. The number of predictors considered at each split and the sample fraction will be included in all combinations. Hyperparameters not included in a given combination will be set to their default value. In addition, a random forest will be fit using the default hyperparameters to establish the baseline. All considered combinations will be used to fit a random forest on each simulated dataset, leading to 459,000 (17*27,000) tuning procedures being performed.

1.4.2 Study 2: Optimisation metric

We will vary the metric to optimise when fitting a random forest using the R package `ranger` via the R package `caret`. We will use grid search (as is standard in this package) to tune the hyperparameters considered optimal in Study 1. 5-fold cross-validation will be used as part of the tuning procedure. The following candidate metrics will be considered:

Table 3: Optimisation metric targets

Metric	Target	Range of possible values
Accuracy	1	[0, 1]
Kappa	1	[0, 1]
Brier score	0	[0, 1]
Logarithmic loss	0	[0, 1]
AUC	1	[0.5, 1]
Calibration intercept	0	$[-\infty, \infty]$
Calibration slope	1	$[0, \infty]$

- Classification accuracy, which measures the proportion of correctly classified observations.
- Cohen's Kappa, which measures the proportion of correctly classified observations while accounting for chance.
- Brier score, which measures the difference between the predicted probability and the true outcome. This can be decomposed into a calibration component and a refinement component, which is related to the AUC. As such, the Brier score can be seen as a composite measire of calibration and discrimination [4].
- Logarithmic loss, which measures the difference between the predicted probability and the true outcome while penalising overconfident misclassifications.
- AUC, which measures how well classes can be differentiated.
- Calibration intercept (if possible to implement), which measures the distance between the average predicted risk and the event rate.
- Calibration slope (if possible to implement), which measures the extent of over- or underestimation.

Target values for each of these metrics are detailed in Table 3.

That is, each dataset will be tuned 7 times, leading to 189,000 tuning procedures.

1.4.3 Study 3: Hyperparameter search algorithm

We will vary the hyperparameter search algorithm when fitting a random forest. We will tune the hyperparameters considered most optimal in Study 1 and optimise the metric considered most optimal in Study 2. 5-fold cross-validation will be used as part of the tuning procedure. The following candidate hyperparameter search algorithms will be considered:

- Model-free search algorithms:

  - Grid search using the R package `caret`,
  - Random search using the R package `caret`,
- Bayesian optimisation: SMAC using the R package `tuneRanger`,
- Multifidelity: Hyperband using the R package `mlr3hyperband`,
- Metaheuristic: genetic algorithm using the R package `GA`.

That is, each dataset will be tuned 5 times, leading to 132,000 tuning procedures.

1.5 Performance measures

For each tuning procedure performed, primary outcomes will be:

- Discrimination (AUC),
- Calibration slope (calculated using [5]),
- Root mean square of the log of the calibration slope over all the runs of a scenario (RMSD(slope)), as used in [6],
- Computational time.

Secondary outcomes will be:

- Calibration intercept,
- Brier score,
- Logarithmic loss,
- Classification accuracy with a threshold of 0.5,
- Kappa.

Model performance metrics will be estimated using the predictions of the model on an independently generated validation set generated under the same data generating mechanisms. For each data simulation scenario and tuning procedure combination, we will compute the average and spread of each of these performance measures, leading to a table of the form of Table 4, Table 5, and Table 6 for Studies 1, 2, and 3, respectively.

We will evaluate and compare performance between hyperparameter combinations, optimisation metrics, and hyperparameter seach algorithms. This will be done using visualisations (e.g., scatterplots with time on the x-axis and performance metrics on the y-axis) and average performances and their spread. We aim to visually assess whether certain hyperparameter combinations, optimisation metrics, or hyperparameter seach algorithms have a notably larger runtime compared to others, for a relatively low increase in performance. The best hyperparameter combination, optimisation metric, and hyperparameter seach algorithm for model performance will be selected in Studies 1, 2, and 3, respectively. Selection will be done considering all primary outcomes.

Table 4: Study 1 outcome table.

Data simulation settings			Hyperparameters tuned	Performance metrics			Time
p	Event fraction	Sample size		AUC	Calibration slope	RMSD(slope)	
8	0.1	$0.5N$	none	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
16	0.1	$0.5N$	none	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
...	...	...	none	...	...	...	...
8	0.1	$0.5N$	mtry + min.node.size	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
16	0.1	$0.5N$	mtry + min.node.size	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
...	...	...	mtry + min.node.size	...	...	...	...
...	...	...	...	...	...	...	...

The final table will have 459 rows.

Table 5: Study 2 outcome table.

Data simulation settings			Optimisation metric	Performance metrics			Time
p	Event fraction	Sample size		AUC	Calibration slope	RMSD(slope)	
8	0.1	$0.5N$	Accuracy	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
16	0.1	$0.5N$	Accuracy	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
...	...	...	Accuracy	...	...	...	...
8	0.1	$0.5N$	Kappa	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
16	0.1	$0.5N$	Kappa	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
...	...	...	Kappa	...	...	...	...
...	...	...	...	...	...	...	...

The final table will have 189 rows.

Table 6: Study 3 outcome table.

Data simulation settings			Hyperparameters search algorithm	Performance metrics			Time
p	Event fraction	Sample size		AUC	Calibration slope	RMSD(slope)	
8	0.1	$0.5N$	Grid search	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
16	0.1	$0.5N$	Grid search	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
...	...	...	Grid search	...	...	...	...
8	0.1	$0.5N$	Random search	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
16	0.1	$0.5N$	Random search	Mean (Variance)	Median (IQR)	NA	Mean (Variance)
...	...	...	Random search	...	...	...	...
...	...	...	...	...	...	...	...

The final table will have 135 rows.

2 Error handling

2.1 Degenerate outcome distributions

The number of datasets with zero events or non-events per simulation scenario will be reported. These datasets will not be used further. If this occurs for a validation dataset, a new validation dataset will be generated to replace it.

2.2 Non-converging calibration slopes

The number of non-converging calibration slopes per data simulation scenario and factor being varied (i.e., hyperparameter combination in study 1, optimisation metric in study 2, hyperparameter search algorithm in study 3) will be reported. Non-converging calibration slopes will be imputed as the highest calibration slope for the given setting, as this would typically occur for severely underfit models.

References

[1] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. "Tunability: Importance of hyperparameters of machine learning algorithms". In: *The Journal of Machine Learning Research* 20.1 (2019). Publisher: JMLR. org, pp. 1934–1965.

[2] Richard D. Riley et al. "Calculating the sample size required for developing a clinical prediction model". en. In: *BMJ* 368 (Mar. 2020). Publisher: British Medical Journal Publishing Group Section: Research Methods & Reporting, p. m441. ISSN: 1756-1833. DOI: 10.1136/bmj.m441. URL: https://www.bmj.com/content/368/bmj.m441 (visited on 10/07/2022).

[3] Erwan Scornet. "Tuning parameters in random forests". en. In: *ESAIM: Proceedings and Surveys* 60 (2017). Publisher: EDP Sciences, pp. 144–162. ISSN: 2267-3059. DOI: 10.1051/proc/201760144. URL: https://www.esaim-proc.org/articles/proc/abs/2017/05/proc186008/proc186008.html (visited on 12/04/2022).

[4] K. Luijken et al. "Impact of predictor measurement heterogeneity across settings on the performance of prediction models: A measurement error perspective". en. In: *Statistics in Medicine* 38.18 (2019). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.10 pp. 3444–3459. ISSN: 1097-0258. DOI: 10.1002/sim.8183. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8183 (visited on 11/29/2022).

[5] benvancalster. *benvancalster/classimb_calibration*. original-date: 2022-02-14T15:52:06Z. Nov. 2022. URL: https://github.com/benvancalster/classimb_calibration/blob/ad521b46b32ec42689a05bc336fb3270a5c1f28e/simulation%20study/Simulation/performance_measures_wo_eci.R (visited on 11/29/2022).

[6] Ben Van Calster et al. "Regression shrinkage methods for clinical prediction models do not guarantee improved performance: Simulation study". en. In: *Statistical Methods in Medical Research* 29.11 (Nov. 2020). Publisher: SAGE Publications Ltd STM, pp. 3166–3178. ISSN: 0962-2802. DOI: 10.1177/0962280220921415. URL: https://doi.org/10.1177/0962280220921415 (visited on 11/29/2022).