Simulation protocol

Judith Neve

November 20, 2022

1 Studies

1.1 Study 1: Hyperparameters to tune

1.1.1 Aims

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 model performance.

1.1.2 Data-generating mechanism

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

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

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`.

normal distribution with parameters detailed in equation 1.

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

Additionally, $0.25p$ interactions will be computed. Interactions will be pairwise and no predictor will appear in more than one interaction term. Then, the binary outcome y_i will be drawn from a Bernoulli distribution conditional on $\mathbf{x}_i$, computed interactions, and the true effects.

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.

True effect estimation True effects will be determined as follows: for each combination k of number of candidate predictor and event fraction, the intercept $\beta_0^{(k)}$, predictor slopes $\boldsymbol{\beta}^{(k)}$, and interaction slopes $\boldsymbol{\gamma}^{(k)}$ will be estimated using a large sample (N = 100,000) approximation. All elements of $\boldsymbol{\beta}^{(k)}$ will be set to be equal to each other and all elements of $\boldsymbol{\gamma}^{(k)}$ will be set to be equal to each other. 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, and ii) the absolute difference between the targeted event fraction and the average predicted probability in the 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 $\boldsymbol{\beta}^{(k)}$, optimise $\boldsymbol{\gamma}^{(k)}$ for a target 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.

The estimated coefficients will be used to generate a validation dataset (N = 1,000,000) for each event fraction and number of candidate predictors combination, on which four things will be checked:

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.

1.1.3 Estimands

The focus of this study is on predictive performance for dichotomous outcome models and necessary computational power.

1.1.4 Methods

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 industry standard) to optimise accuracy (as is the default in `caret`). 5-fold cross-validation will be used as part of the tuning procedure.

All combinations of the following hyperparameters will be considered:

- number of trees,
- replace,
- respect unordered factors,
- minimum node size,
- split rule.

This leads to 32 different combinations. The number of predictors considered at each split and the sample fraction will be included in all combinations. 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 891,000 (33*27,000) tuning procedures being performed.

1.1.5 Performance measures

For each tuning procedure performed, we look at:

- Discrimination (AUC),
- Calibration:
  - Calibration slope,
  - Calibration in the large.
- Computational time.

Model performance metrics will be estimated using the predictions of the model on their appropriate validation dataset. For each data simulation scenario and tuning procedure combination, we will compute the mean of each of these performance measures, leading to a table of the form of Table 2.

Model performance will be evaluated in a single metric computed using formula 2.

$$\text{AUC} * (1 - \sqrt{ln(\text{calibration slope})^2}) \tag{2}$$

Table 2: Study 1 outcome dataset.

Data simulation settings			Hyperparameters tuned	Performance metrics			Time
p	Event fraction	Sample size		AUC	Calibration slope	CIL	
8	0.1	$0.5N$	none	NA	NA	NA	NA
16	0.1	$0.5N$	none	NA	NA	NA	NA
...	...	...	none	...	...	...	...
8	0.1	$0.5N$	mtry + sample fraction	NA	NA	NA	NA
16	0.1	$0.5N$	mtry + sample fraction	NA	NA	NA	NA
...	...	...	mtry + sample fraction	...	...	...	...
...	...	...	...	...	...	...	...

The final dataset will have 891 rows.

This performance metric will then be used to determine the best combination of hyperparameters. We will check whether all performance metrics are above the performance metrics for the untuned model. If this is not the case, the second best hyperparameter combination will be considered.
Additionally, we will create a scatterplot with time on the x-axis, this metric on the y-axis and points corresponding to the chosen hyperparameter combination plotted in a different colour. The goal of this is to visually assess whether the chosen combination has an abnormally large runtime compared to others, for a relatively low increase in performance. If this is the case, the second best hyperparameter combination will be considered.

1.2 Study 2: Optimisation metric

1.2.1 Aims

This study aims to identify the metric to optimise in the tuning procedure which leads to the best model performance.

1.2.2 Data-generating mechanism

Population New datasets generated following the procedure in study 1.

True effect estimation Effects estimated in study 1 will be used.

1.2.3 Estimands

The focus of this study is on predictive performance for dichotomous outcome models. Computational power is a secondary focus.

1.2.4 Methods

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 the industry standard) for the hyperparameters selected from study 1. 5-fold cross-validation will be used as part of the tuning procedure. The following candidate metrics will be considered:

Table 3: Study 2 outcome dataset.

Data simulation settings			Optimisation metric	Performance metrics			Time
p	Event fraction	Sample size		AUC	Calibration slope	CIL	
8	0.1	$0.5N$	Accuracy	NA	NA	NA	NA
16	0.1	$0.5N$	Accuracy	NA	NA	NA	NA
...	...	...	Accuracy	...	...	...	...
8	0.1	$0.5N$	Kappa	NA	NA	NA	NA
16	0.1	$0.5N$	Kappa	NA	NA	NA	NA
...	...	...	Kappa	...	...	...	...
...	...	...	...	...	...	...	...

The final dataset will have 162 rows.

- Accuracy,
- Kappa,
- Brier score,
- AUC,
- Logarithmic loss,
- Calibration in the large (if possible to implement).

That is, each dataset will be tuned 6 times, leading to 162,000 tuning procedures.

1.2.5 Performance measures

For each tuning procedure performed, we look at:

- Discrimination (AUC),
- Calibration:
  - Calibration slope,
  - Calibration in the large.
- Computational time.

Model performance metrics will be estimated using the predictions of the model on their appropriate validation dataset. For each data simulation scenario and optimisation metric combination, we will compute the mean of each of these performance measures, leading to a table of the form of Table 3.

Model performance will be evaluated in a single metric computed as shown in formula 2. This metric will then be used to determine the best optimisation metric.
Additionally, we will create a scatterplot with time on the x-axis, this metric on the y-axis and points colour-coded according to optimisation metrics. The goal of this is to visually assess whether there seem to be substantial differences in required computational power between the optimisation metrics.

1.3 Study 3: Hyperparameter search algorithm

1.3.1 Aims

This study aims to identify the hyperparameter search algorithm which leads to the best model performance, with a consideration for required computational power.

1.3.2 Data-generating mechanism

Population New datasets generated following the procedure in study 1.

True effect estimation Effects estimated in study 1 will be used.

1.3.3 Estimands

The focus of this study is on predictive performance for dichotomous outcome models and required computational power for fitting these models.

1.3.4 Methods

We will vary the hyperparameter search algorithm when fitting a random forest. We will tune the hyperparameters selected from study 1 and optimise the metric selected from 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.

1.3.5 Performance measures

For each tuning procedure performed, we look at:

- Discrimination (AUC),
- Calibration:
  - Calibration slope,

Table 4: Study 3 outcome dataset.

Data simulation settings			Hyperparameters search algorithm	Performance metrics			Time
p	Event fraction	Sample size		AUC	Calibration slope	CIL	
8	0.1	$0.5N$	Grid search	NA	NA	NA	NA
16	0.1	$0.5N$	Grid search	NA	NA	NA	NA
...	...	...	Grid search	...	...	...	...
8	0.1	$0.5N$	Random search	NA	NA	NA	NA
16	0.1	$0.5N$	Random search	NA	NA	NA	NA
...	...	...	Random search	...	...	...	...
...	...	...	...	...	...	...	...

The final dataset will have 135 rows.

  - Calibration in the large.

- Computational time.

Model performance metrics will be estimated using the predictions of the model on their appropriate validation dataset. For each data simulation scenario and tuning procedure combination, we will compute the mean of each of these performance measures, leading to a table of the form of Table 4.

Model performance will be evaluated in a single metric computed using formula 2. This metric will then be used to determine the best hyperparameter search algorithm.
Additionally, we will create a scatterplot with time on the x-axis, this metric on the y-axis and points corresponding to the chosen hyperparameter combination plotted in a different colour. The goal of this is to visually assess whether the chosen search algorithm has an abnormally large runtime compared to others, for a relatively low increase in model performance. If this is the case, the second best hyperparameter combination will be considered.

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).