NTNU | Norwegian University of Science and Technology

DESIGN OF EXPERIMENTS AND RESPONSE SURFACE METHODOLOGY TO TUNE MACHINE LEARNING HYPERPARAMETERS, WITH A RANDOM FOREST CASE-STUDY

Article presentation in MA8701 by Javier, Håkon and Yngvild

Introduction - Objective & previous approaches

We want to apply a machine learning algorithm to perform regression/classification

Introduction - Objective & previous approaches

We want to apply a machine learning algorithm to perform regression/classification

Objective: find hyperparameter configuration that optimizes our performance metric

Introduction - Objective & previous approaches

We want to apply a machine learning algorithm to perform regression/classification
Objective: find hyperparameter configuration that optimizes our performance metric
Common approaches:

- Stochastic gradient descent
- Gaussian process-based Bayesian optimization
- Random search

Introduction - Idea in this paper

1. Find most important hyperparameters (factors) in the random forest algorithm using design of experiments (DOE)

Introduction - Idea in this paper

1. Find most important hyperparameters (factors) in the random forest algorithm using design of experiments (DOE)

2. Apply response surface methodology (RSM) on the parameters chosen in step 1

Background - Design of experiments (DOE)

Experiment: series of systematic tests which attempt to find the factors which have the largest effect on a response variable.

Main Objective: Optimize the response variable.

This involves:

- Careful selection of variables
- Ranges of variables
- Number of experiments and their order

Background - Design of experiments (DOE)

Traditionally DOE has been performed by changing a factor a time.

Inefficient! Misses information about interactions. Usually overlooked in hyperparameter tuning efforts.

Background - Design of experiments (DOE)

A response variable may be impacted by controllable and uncontrollable factors.

- **Controllable factor:** The experimenter can freely alter its levels.
- **Uncontrollable factor:** Variables that are not controlled by the experimenter, but can be monitored and even included in the model.

Background - Design of experiments (DOE)

Principles of DOE:

1. **Randomization:** experiments should be run in a random order to prevent external factor from affecting results.

Background - Design of experiments (DOE)

Principles of DOE:

1. **Randomization:** experiments should be run in a random order to prevent external factor from affecting results.
2. **Replication:** allows calculation of internal s.e

Background - Design of experiments (DOE)

Principles of DOE:

1. **Randomization:** experiments should be run in a random order to prevent external factor from affecting results.
2. **Replication:** allows calculation of internal s.e
3. **Blocking:** can reduce variability

Background - Design of experiments (DOE)

Principles of DOE:

1. **Randomization:** experiments should be run in a random order to prevent external factor from affecting results.
2. **Replication:** allows calculation of internal s.e
3. **Blocking:** can reduce variability

Background - Design of experiments (DOE)

Two level factorial design (2^k):

- Most basic type of experiment.
- k factors at two levels: low and high.
- Regression model:

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i<j} \beta_{ij} x_i x_j + \varepsilon$$

where $\beta_i, i = 1, ..k$ are main effects and $\beta_{ij}, j = 2, ..., k$ are interaction terms.

As k increases, the number of runs increases exponentially.

Idea: use a fractional DOE

Background - Design of experiments (DOE)

Fractional Factorial DOE (2^{k-p}):

1. Fewer runs are needed 2^{k-p}.
2. Trade-off: loss of accuracy due to fewer df to evaluate each factor and every possible interaction.
3. Powerful screening methods. Usually done at the beginning of experiment to see which factors are important.

Background - Design of experiments (DOE)

3 unique characteristics that make them highly efficient:

1. **Sparsity of effects principle:** only a small number of effects are significant and the final model is composed of low order terms.

Background - Design of experiments (DOE)

3 unique characteristics that make them highly efficient:

1. **Sparsity of effects principle:** only a small number of effects are significant and the final model is composed of low order terms.
2. **Projection property:** a design can be projected into a lower dimension using a subset of factors.

Background - Design of experiments (DOE)

3 unique characteristics that make them highly efficient:

1. **Sparsity of effects principle:** only a small number of effects are significant and the final model is composed of low order terms.
2. **Projection property:** a design can be projected into a lower dimension using a subset of factors.
3. **Fold over:** FFDOE can be combined to form designs of higher resolution Helps in isolating main effects.

Background - Design of experiments (DOE)

3 unique characteristics that make them highly efficient:

1. **Sparsity of effects principle:** only a small number of effects are significant and the final model is composed of low order terms.
2. **Projection property:** a design can be projected into a lower dimension using a subset of factors.
3. **Fold over:** FFDOE can be combined to form designs of higher resolution Helps in isolating main effects.

Serious disadvantage of FFDOE: unable to detect quadratic effects.

Background - Design of experiments (DOE)

Solution: add a third level of **center points** to one or more factors in addition to the two levels in a 2^k DOE.

- Center points are coded as 0.
- Does not impact the effect estimates $\hat{\beta}_j, j \geq 1$ and $\hat{\beta}_0$ becomes the average.
- Adding center points helps us test lack of fit, since it is expected that

$$\bar{y}_f - \bar{y}_c \approx 0,$$

  where $\bar{y}_f$ is the mean of the factorial design and $\bar{y}_c$ is the mean of center points.
- Additionally we can estimate the pure error at the center point and partition

$$SSE = SS_{PE} + SS_{LOF}.$$

Background - Response Surface Methodology

RSM: Procedure used to model a surface using statistical techniques for the purpose of optimizing a response.

Objective: Find value of x that maximizes response y, with

$$y = f(x) + \varepsilon,$$

where ε is the error and the response surface is $\eta = f(x)$.

Challenge: a priori f is an unknown function.

Background - Response Surface Methodology

Methodology: find a model which fits the relationship between the predictors and the response using a polynomial function.

Popular choices:

- **First-order model:**

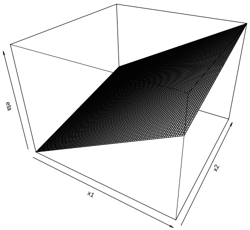
$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \varepsilon$$

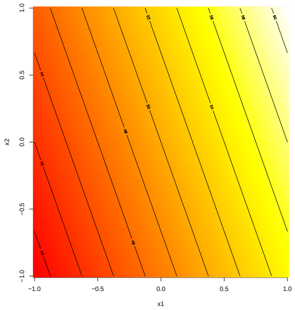
- **Second-order model:**

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{jj} x_j^2 + \sum_{i<j} \beta_{ij} x_i x_j + \varepsilon$$

Background - Response Surface Methodology

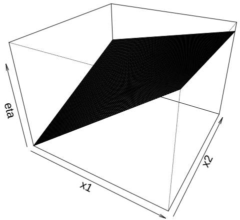
Main effects

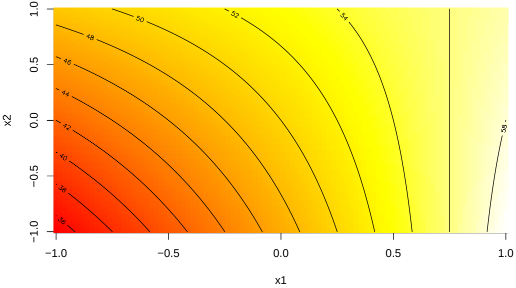




Background - Response Surface Methodology

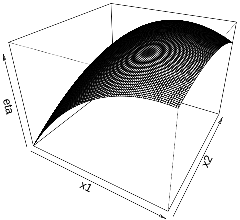
Interaction

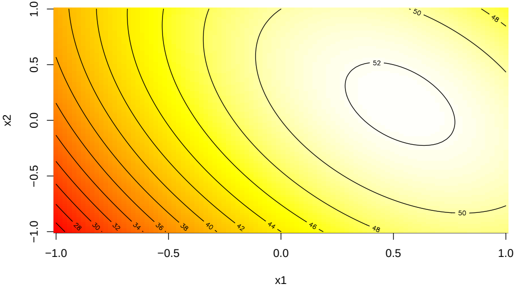




Background - Response Surface Methodology

Quadratic





Background - Response Surface Methodology

- RSM is **sequential procedure** where at each step, we move in a direction of improvement for our objective.

Background - Response Surface Methodology

- RSM is **sequential procedure** where at each step, we move in a direction of improvement for our objective.
- **Steepest ascent** or ridge analysis is used to move to optimal region or the response surface.

Background - Response Surface Methodology

- RSM is **sequential procedure** where at each step, we move in a direction of improvement for our objective.
- **Steepest ascent** or ridge analysis is used to move to optimal region or the response surface.
- The procedure is repeated until no more improvements are found in a local neighborhood.

Background - Response Surface Methodology

- RSM is **sequential procedure** where at each step, we move in a direction of improvement for our objective.
- **Steepest ascent** or ridge analysis is used to move to optimal region or the response surface.
- The procedure is repeated until no more improvements are found in a local neighborhood.

Most popular RSM designs: Central Composite designs (CCD) , Box-Behnken (BBD)

Background - Performance metrics

Balanced accuracy (BACC):

$$BACC = (TPR + TNR)/2, \tag{1}$$

where $TPR = TP/(TP + FN)$ and $TNR = TN/(TN + FP)$.
Good metric for highly unbalanced data

Background - Random forest

Bagging: create B bootstrap samples and fit a decision tree to each sample

Background - Random forest

Bagging: create B bootstrap samples and fit a decision tree to each sample
Random forest: in each split, we are only allowed to consider m of the p predictors

Background - Random forest

Bagging: create B bootstrap samples and fit a decision tree to each sample
Random forest: in each split, we are only allowed to consider m of the p predictors
Use fully-grown trees rather than pruned ones
$\implies$ Less correlated

Background - The `RandomForest` package in R

Hyperparameters in `RandomForest`:

1. ntree: number of trees to grow

Background - The `RandomForest` package in R

Hyperparameters in `RandomForest`:

1. ntree: number of trees to grow
2. mtry: number of predictors m allowed to be considered at each split

Background - The `RandomForest` package in R

Hyperparameters in `RandomForest`:

1. ntree: number of trees to grow
2. mtry: number of predictors m allowed to be considered at each split
3. replace: should sampling be done with our without replacement?

Background - The `RandomForest` package in R

Hyperparameters in `RandomForest`:

1. ntree: number of trees to grow
2. mtry: number of predictors m allowed to be considered at each split
3. replace: should sampling be done with our without replacement?
4. nodesize: minimum size of leaf nodes

Background - The `RandomForest` package in R

Hyperparameters in `RandomForest`:

1. ntree: number of trees to grow
2. mtry: number of predictors m allowed to be considered at each split
3. replace: should sampling be done with our without replacement?
4. nodesize: minimum size of leaf nodes
5. classwt: prior probability for each of the classes

Background - The `RandomForest` package in R

Hyperparameters in `RandomForest`:

1. ntree: number of trees to grow
2. mtry: number of predictors m allowed to be considered at each split
3. replace: should sampling be done with our without replacement?
4. nodesize: minimum size of leaf nodes
5. classwt: prior probability for each of the classes
6. cutoff: threshold for binary classification

Background - The `RandomForest` package in R

Hyperparameters in `RandomForest`:

1. ntree: number of trees to grow
2. mtry: number of predictors m allowed to be considered at each split
3. replace: should sampling be done with our without replacement?
4. nodesize: minimum size of leaf nodes
5. classwt: prior probability for each of the classes
6. cutoff: threshold for binary classification
7. maxnodes: maximum number of leaf nodes a tree can have

Experiments - The dataset

Aim: classifying whether a person makes over 50 000 USD per year
32561 observations, 14 covariates
Some of the covariates:

1. age
2. marital status
3. race
4. sex
5. education

Experiments - General procedure for hypertuning using DOE and RSM

Procedure

1. Choose a machine learning algorithm and decide on the response variable to tune (accuracy, TPR, F1-score, etc.)

Experiments - General procedure for hypertuning using DOE and RSM

Procedure

1. Choose a machine learning algorithm and decide on the response variable to tune (accuracy, TPR, F1-score, etc.)
2. Select the hyperparameters to tune as well as their ranges

Experiments - General procedure for hypertuning using DOE and RSM

Procedure

1. Choose a machine learning algorithm and decide on the response variable to tune (accuracy, TPR, F1-score, etc.)
2. Select the hyperparameters to tune as well as their ranges
3. Perform a screening design and identify the important hyperparameters

Experiments - General procedure for hypertuning using DOE and RSM

Procedure

1. Choose a machine learning algorithm and decide on the response variable to tune (accuracy, TPR, F1-score, etc.)
2. Select the hyperparameters to tune as well as their ranges
3. Perform a screening design and identify the important hyperparameters
4. Reduce the model and, depending on the number of experiments that are feasible to run, perform either a full or fractional 2k factorial design

Experiments - General procedure for hypertuning using DOE and RSM

Procedure

1. Choose a machine learning algorithm and decide on the response variable to tune (accuracy, TPR, F1-score, etc.)
2. Select the hyperparameters to tune as well as their ranges
3. Perform a screening design and identify the important hyperparameters
4. Reduce the model and, depending on the number of experiments that are feasible to run, perform either a full or fractional 2k factorial design
5. Fit a second-order model using RSM (CCD, BBD), selecting the hyperparameter configuration with the best performance from the previous step as the center of the design

Experiments - General procedure for hypertuning using DOE and RSM

Procedure

1. Choose a machine learning algorithm and decide on the response variable to tune (accuracy, TPR, F1-score, etc.)
2. Select the hyperparameters to tune as well as their ranges
3. Perform a screening design and identify the important hyperparameters
4. Reduce the model and, depending on the number of experiments that are feasible to run, perform either a full or fractional 2k factorial design
5. Fit a second-order model using RSM (CCD, BBD), selecting the hyperparameter configuration with the best performance from the previous step as the center of the design
6. Recursively optimize the second-order model until the change in the response is $\leq \epsilon$.

Experiments - Comments to the procedure

- Throughout each of these steps, the response variable should be estimated using n-fold cross-validation.
- The result of the procedure will be compared to the default settings
- The data set is small enough to accommodate a full factorial as the first run, but they choose to pretend that initial screening is needed
- The initial screening is performed using a 2^{7-2} design, so some two-factor interactions are confounded

Experiments - Initial levels for screening

Table: Factors and levels in the initial screening

Factor	Low factor level (-)	High factor level (+)
ntree	100	500
mtry	2	4
replace	FALSE	TRUE
nodesize	1	3256
classwt	1	10
cutoff	0.2	0.8
maxnodes	5	NULL

Experiments - Analysis of first screening

Coefficients	Estimate	Std. Error	t-value	$P(>\|t\|)$
(Intercept)	0.3458	0.0043	80.503	2.47E-10 ***
ntree	0.0029	0.0043	0.684	0.5193
mtry	–0.0069	0.0043	–1.614	0.1578
replace	–0.0253	0.0043	–5.879	0.0011 **
nodesize	0.0435	0.0043	10.132	5.37E-05 ***
classwt	–0.1364	0.0043	–31.766	6.47E-08 ***
cutoff	0.0475	0.0043	11.07	3.24E-05 ***
maxnodes	–0.0593	0.0043	–13.816	8.95E-06 ***
ntree:mtry	–0.0371	0.0043	–8.636	0.0001 ***
ntree:replace	0.0003	0.0043	0.085	0.9357

- Confounded effects significant, need follow-up. Use fold over design.

Experiments - Analysis of second screening

Coefficients	Estimate	Std. Error	t-value	$P(>\|t\|)$
(Intercept)	5.92E-01	7.82E-03	75.777	2E-16
ntree	–9.07E-04	7.82E-03	–0.116	0.9082
mtry	5.36E-03	7.82E-03	0.686	0.4975
replace	1.61E-03	7.82E-03	0.206	0.8377
nodesize	–6.41E-03	7.82E-03	–0.821	0.4174
classwt	–1.42E-02	7.82E-03	–1.818	0.0777 +
cutoff	–3.06E-03	7.82E-03	–0.391	0.6978
maxnodes	1.39E-02	7.82E-03	1.782	0.0834 +
ntree:mtry	4.29E-04	7.82E-03	0.055	0.9566
ntree:replace	–3.16E-03	7.82E-03	–0.405	0.6882

- Significant two-factor interactions: The hierarchy and heredity dilemma

Main results - Initial screening

- ntree not significant - saving computations by setting it low
- Note: A hyperparameter not being significant in this particular case can matter in other settings
- Having identified the active factors, a full factorial experiment was conducted
- Results analyzed, maxnodes removed, new full factorial with factors nodesize, classwt and cutoff

Main results - RSM for optimization

- Having completed the screening phase, it was time to optimize
- Used Box Behnken design, suited for fitting second-order models (several levels for each factor)
- Fitted model, found the significant terms, fitted reduced model
- Steepest ascent, but not outside the experimental region
- New experiment, new model and new steepest ascent
- Satisfying results - 0.81 in BACC compared to the default 0.64

Discussion and conclusion - part 1

- Saving computations by using low levels of hyperparameters that are not significant
- Some parameter can compensate for each other
- Method allows us to understand which hyperparameters matter and how they impact the result - but the spesifics do not necessarily generalize
- Convexity unrealistic - probably found local maximum

Discussion and conclusion - part 2: Our comments

- Advantages of the method: Can save computation and gain information about which hyperparameters matter
- Disadvantage: Not possible to use this if very many hyperparameters must be tuned. Requires a lot of domain knowledge. Should probably be automated to achieve popularity
- Would have been interesting: Comparison with grid search and Bayesian optimization
- More information about computational demands
- Confidence intervals for BACC

Thank you for your attention

NTNU | Norwegian University of Science and Technology