A Minimal Book Example

Yihui Xie

2020-02-18

Contents

Chapter 1

thesis Ferdinands

Goal: evaluate performance of different models of the ASReview tool.

1.0.1 Stage 1: hyperparameter optimization

We are testing 5 models on 5 different datasets.

Every model has its own set of hyperparameters. The hyperparameters are optimized on the 5 datasets in three different ways:

- 1 on 1: maximum performance
- 4 on 1: cross-validation
- 5 on 1: more data = more better?

This results $(5 + 5 + 1) * 5$ sets of hyperparameters.

1.0.2 Stage 2: simulation

for every for every model (5), for every dataset (5) and for every set of optimized hyperparameters (3), a simulation study is performed. From these $5 * 5 * 3 = 75$ simulation studies, performance of the different models is evaluated.

1.1 Scripts

- simulatie:

☐ script die optimalisatie genereert (config files.)
☐ script die `.ini`'s voor simulatie genereert.

Chapter 2

Analysis strategy

2.1 Datasets

- ptsd
- ace
- hall
- nagtegaal - van PhD van Lars
- medische van Jan

2.2 Models

(input table here?)

- Naive Bayes
- Random Forests
- Support Vecor Machine (doc2vec)
- Logistic Regression
- Dense Neural Network (doc2vec)

The other parameters remain fixed over the 5 models:

- Feature extraction = tf-idf
- Query Strategy = max
- Balance Strategy = triple
- n_instances=10 (number of papers each query)
- n_prior_included = 5
- n_prior_excluded = 5
- mix_ratio = 0.95 (95% max, 5% random)

2.3 Hyperparameters

Chapter 3

My first simulation

3.1 Methods

A simulation study was performed on the Nagtegaal dataset, using a model with the following configurations:

- Model = Naive Bayes
- Query Strategy = max_random
- Balance Strategy = Double
- n_instances=10 (number of papers each query)
- n_papers=2000 (shouldn't I do all?)
- n_prior_included = 5
- n_prior_excluded = 5
- mix_ratio = 0.95 (95% max, 5% random)

Hyperparameters		default	optimized
Model			
	alpha	3.822	3.511844
Balance			
	a	2.155	0.254892
	alpha	0.94	1.459081
	b	0.789	0.394437
Feature			
	ngram_max	1	2
	split_ta	0	1

For the sake of evaluating the optimized hyperparameters, two simulations of five runs each were compared: one with default hyperparameters and with optimized hyperparameters.

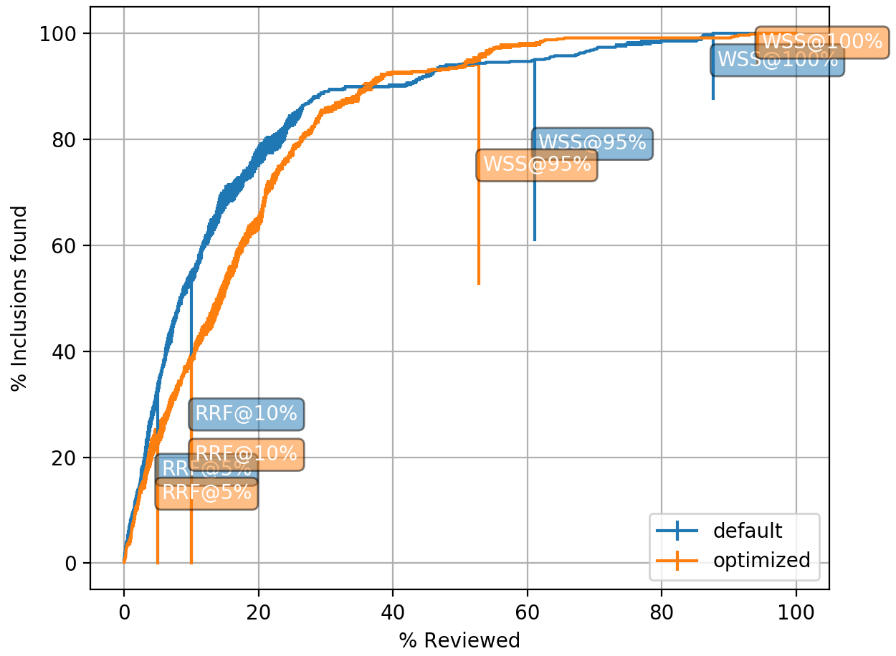
3.2 Results

Explanation of the plots come from the `asreview-visualization` repository. The optimized hyperparameters do not perform better than the default ones, this is probably due to the fact that the default hyperparameters have already been optimized in the past. It is therefore to know for which models this has already been done and which not!

3.2.0.1 Inclusions

This figure shows the number/percentage of included papers found as a function of the number/percentage of papers reviewed. Initial included/excluded papers are subtracted so that the line always starts at (0,0).

The quicker the line goes to a 100%, the better the performance.

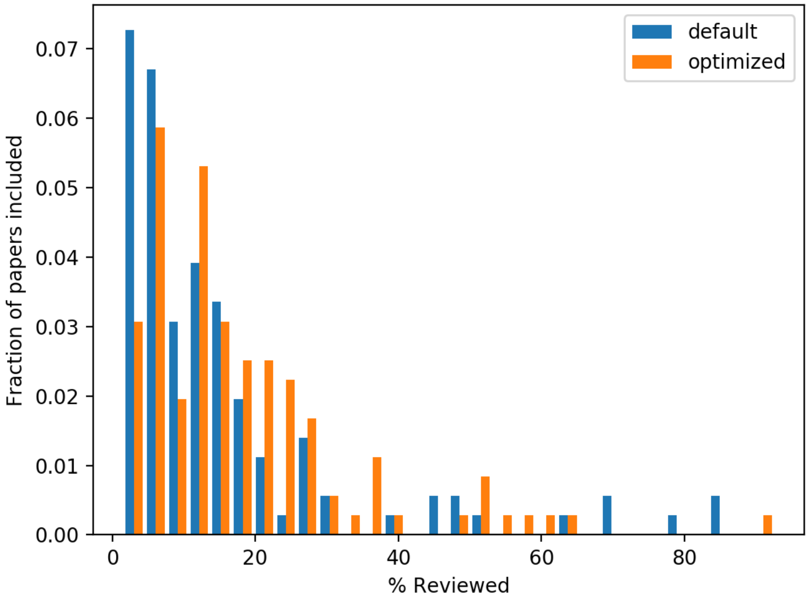


In the beginning, the model with default parameters finds inclusions quicker than the model with optimized hyperparamters. Only after reviewing 50% of the papers, the optimized hyperparameters outperform the default ones.

3.2.0.2 Discovery

This figure shows the distribution of the number of papers that have to be read before discovering each inclusion. Not every paper is equally hard to find.

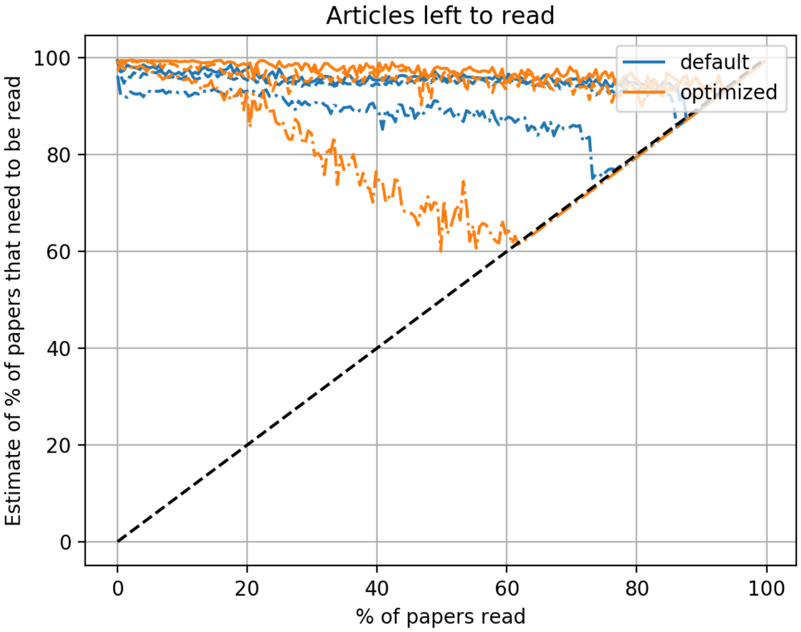
The closer to the left, the better.



3.2.0.3 Limits

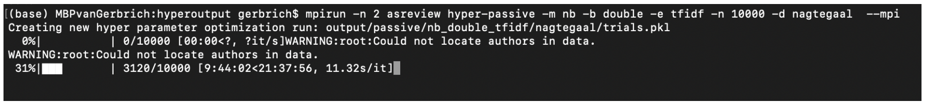
This figure shows how many papers need to be read with a given criterion. A criterion is expressed as "after reading y % of the papers, at most an average of z included papers have been not been seen by the reviewer, if he is using max sampling.". Here, y is shown on the y-axis, while three values of z are plotted as three different lines with the same color. The three values for z are 0.1, 0.5 and 2.0.

The quicker the lines touch the black (`y=x`) line, the better.



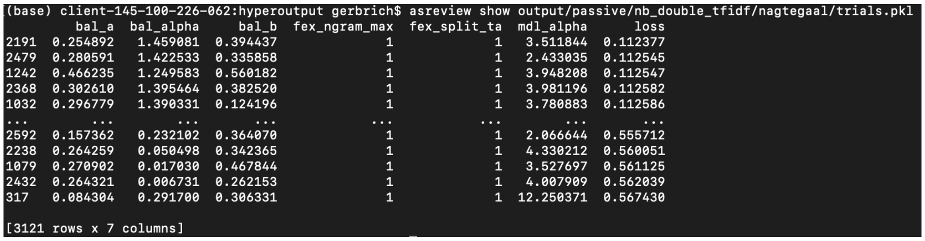
3.3 Console output

3120 iterations ran overnight.



```
[(base) MBPvanGerbrich:hyperoutput gerbrich$ mpirun -n 2 asreview hyper-passive -m nb -b double -e tfidf -n 10000 -d nagtegaal  --mpi
 Creating new hyper parameter optimization run: output/passive/nb_double_tfidf/nagtegaal/trials.pkl
   0%|          | 0/10000 [00:00<?, ?it/s]WARNING:root:Could not locate authors in data.
 WARNING:root:Could not locate authors in data.
  31%|███       | 3120/10000 [9:44:02<21:37:56, 11.32s/it]█
```

2191 was best performing with a loss of 0.1124



```
(base) client-145-100-226-062:hyperoutput gerbrich$ asreview show output/passive/nb_double_tfidf/nagtegaal/trials.pkl
         bal_a  bal_alpha     bal_b  fex_ngram_max  fex_split_ta  mdl_alpha      loss
2191  0.254892   1.459081  0.394437              1             1   3.511844  0.112377
2479  0.280591   1.422533  0.335858              1             1   2.433035  0.112545
1242  0.466235   1.249583  0.560182              1             1   3.948208  0.112547
2368  0.302610   1.395464  0.382520              1             1   3.981196  0.112582
1032  0.296779   1.390331  0.124196              1             1   3.780883  0.112586
...        ...        ...       ...            ...           ...        ...       ...
2592  0.157362   0.232102  0.364070              1             1   2.066644  0.555712
2238  0.264259   0.050498  0.342365              1             1   4.330212  0.560051
1079  0.270902   0.017030  0.467844              1             1   3.527697  0.561125
2432  0.264321   0.006731  0.262153              1             1   4.007909  0.562039
317   0.084304   0.291700  0.306331              1             1  12.250371  0.567430

[3121 rows x 7 columns]
```

3.4 Notes

- Optimizing the hyperparameters with the Nagtegaal set and then running a simulation on the Nagtegaal dataset is using the data twice. This leads to overfitting. A next approach could be to perform some way of cross-validation, e.g. split the datasets in train and test datasets.

- There are two main optimization modes: passive and active learning. The first is used here and is relatively fast, the second is more computationally expensive.

- Of primary interest is the comparison of different model configurations in predictive performance. A simulation study can be performed with all possible configurations using the default hyperparameters. The results could be used to select model configurations that could possibly benefit from hyperparameter sreening.

- Second, we could investigate how much is to gain in predictive performance from optimizing the hyperparameters. For this, some cross-validation strategy should be used. Optimization can consist of two steps: first, optimization through passive learning can be performed, from which the best performing models can be selected for the second step: optimization through passive learning.

3.4.1 Possible Research Questions

- Which model configurations have good predictive performance?

    - for what kind of data sets and under which circumstances?

- Does optimization of hyperparameters lead to substantial gain in predictive performance?
    - How much and why?
    - How do the hyperparameters relate to one another?
    - What is the optimal way to tune the hyperparameters?
        - to determine by cross-validation
        - for example: optimize over a large number of datasets? or a different strategy?
- ...
- ...

Appendix A

List of definitions