Hyperparameter tuning

Understanding validation set overfitting when tuning your hyperparameters
Team - P37

Mart Mägi
MSc student in Data Science

Musa Salamov
MA student in Innovation and Technology Management

Kaja Jakobson
MSc student in Conversion Master in IT

Project owner

Viacheslav Komisarenko
Ph.D. student and Junior Research Fellow in Machine Learning at UT
The problem

Hyperparameter tuning:

- goal is to achieve an optimal model
- involves dozens of evaluations on the validation set:
  - noise from random data gets included in model
  - resulting in the best hyperparameters being overfitted on the validation set

Explore the limitation of validation set usage for hyperparameter tuning
Initial expectations

Theoretical background
Hyperparameter optimization can lead to overfitting;

Empirical study
Setting test environment; Performing hyperparameter optimization; Interpreting results;

Outcomes
Better understanding of the overfitting which is caused by hyperparameter tuning; possible strategies that mitigate/overcome the problem
Empirical study

The test environment

Artificially generated data
Number of futures: 10 vs 20; Number of instances: 1000 vs 10000

Hyperparameter tuning
Automated: GridSearchCV
Manually: for 2 hyperparameters

Classifier
Logistic regression
Hyperparameters for LR

1. **C**: float, default: 1.0 : Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.
2. **tol**: float, default: 1e-4 : Tolerance for stopping criteria. This tells the algorithm to stop searching for a minimum (or maximum) once some tolerance is achieved, i.e. once it is close enough.
4. **max_iter**: int, default: 100 : Useful only for the newton-cg, sag and lbfgs solvers. Maximum number of iterations taken for the solvers to converge.
Solver selection

<table>
<thead>
<tr>
<th>Sl</th>
<th>C</th>
<th>Train_acc_liblinear</th>
<th>Val_acc_liblinear</th>
<th>Build_time_liblinear</th>
<th>Train_acc_newton-cg</th>
<th>Val_acc_newton-cg</th>
<th>Build_time_newton-cg</th>
<th>Train_acc_lbfgs</th>
<th>Val_acc_lbfgs</th>
<th>Build_time_lbfgs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1</td>
<td>0.001</td>
<td>82.835821</td>
<td>83.030303</td>
<td>0.001758</td>
<td>69.402965</td>
<td>69.090909</td>
<td>0.006571</td>
<td>69.402985</td>
<td>69.090909</td>
</tr>
<tr>
<td>2.0</td>
<td>2</td>
<td>0.006</td>
<td>82.686567</td>
<td>83.636364</td>
<td>0.001376</td>
<td>76.119403</td>
<td>76.969697</td>
<td>0.006133</td>
<td>76.119403</td>
<td>76.969697</td>
</tr>
<tr>
<td>3.0</td>
<td>3</td>
<td>0.011</td>
<td>82.696567</td>
<td>83.333333</td>
<td>0.001390</td>
<td>79.104478</td>
<td>79.686970</td>
<td>0.005732</td>
<td>79.104478</td>
<td>79.686970</td>
</tr>
<tr>
<td>4.0</td>
<td>4</td>
<td>0.016</td>
<td>82.950755</td>
<td>83.636364</td>
<td>0.001373</td>
<td>80.89522</td>
<td>81.818182</td>
<td>0.006945</td>
<td>80.89522</td>
<td>81.818182</td>
</tr>
<tr>
<td>5.0</td>
<td>5</td>
<td>0.021</td>
<td>83.293582</td>
<td>83.333333</td>
<td>0.001444</td>
<td>81.194030</td>
<td>83.030303</td>
<td>0.007570</td>
<td>81.194030</td>
<td>83.030303</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>996.0</td>
<td>996</td>
<td>4.976</td>
<td>83.134328</td>
<td>84.242424</td>
<td>0.001697</td>
<td>82.95075</td>
<td>84.545455</td>
<td>0.007992</td>
<td>82.95075</td>
<td>84.545455</td>
</tr>
<tr>
<td>997.0</td>
<td>997</td>
<td>4.981</td>
<td>83.134328</td>
<td>84.242424</td>
<td>0.001753</td>
<td>82.95075</td>
<td>84.545455</td>
<td>0.008276</td>
<td>82.95075</td>
<td>84.545455</td>
</tr>
<tr>
<td>998.0</td>
<td>998</td>
<td>4.986</td>
<td>83.134328</td>
<td>84.242424</td>
<td>0.001744</td>
<td>82.95075</td>
<td>84.545455</td>
<td>0.007393</td>
<td>82.95075</td>
<td>84.545455</td>
</tr>
<tr>
<td>999.0</td>
<td>999</td>
<td>4.991</td>
<td>83.134328</td>
<td>84.242424</td>
<td>0.001721</td>
<td>82.95075</td>
<td>84.545455</td>
<td>0.007218</td>
<td>82.95075</td>
<td>84.545455</td>
</tr>
<tr>
<td>1000.0</td>
<td>1000</td>
<td>4.996</td>
<td>83.134328</td>
<td>84.242424</td>
<td>0.001749</td>
<td>82.95075</td>
<td>84.545455</td>
<td>0.007863</td>
<td>82.95075</td>
<td>84.545455</td>
</tr>
</tbody>
</table>

1000 rows x 17 columns

- Best result: liblinear
Relatively small data

- Number of instances: 1000
- Number of features: 10

- Initial val_acc: 73.33%
- Max val_acc: 88.18%
Relatively big data

- Number of instances: 10000
- Number of features: 20

- Initial val_acc: 90%
- Max val_acc: 90.03%

<table>
<thead>
<tr>
<th>Sl</th>
<th>C</th>
<th>Train_acc</th>
<th>Val_acc</th>
<th>Build_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.001</td>
<td>89.940299</td>
<td>90.00000</td>
<td>0.020243</td>
</tr>
<tr>
<td>2.0</td>
<td>2.006</td>
<td>90.000000</td>
<td>90.00000</td>
<td>0.023894</td>
</tr>
<tr>
<td>3.0</td>
<td>3.011</td>
<td>90.029851</td>
<td>89.939394</td>
<td>0.023087</td>
</tr>
<tr>
<td>4.0</td>
<td>4.016</td>
<td>90.000000</td>
<td>90.00000</td>
<td>0.015033</td>
</tr>
<tr>
<td>5.0</td>
<td>5.021</td>
<td>90.000000</td>
<td>89.969697</td>
<td>0.028266</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>996.0</td>
<td>996</td>
<td>4.976</td>
<td>90.089552</td>
<td>90.030303</td>
</tr>
<tr>
<td>997.0</td>
<td>997</td>
<td>4.961</td>
<td>90.089552</td>
<td>90.030303</td>
</tr>
<tr>
<td>998.0</td>
<td>998</td>
<td>4.966</td>
<td>90.089552</td>
<td>90.030303</td>
</tr>
<tr>
<td>999.0</td>
<td>999</td>
<td>4.991</td>
<td>90.089552</td>
<td>90.030303</td>
</tr>
<tr>
<td>1000.0</td>
<td>1000</td>
<td>4.996</td>
<td>90.089552</td>
<td>90.030303</td>
</tr>
</tbody>
</table>

1000 rows x 5 columns
Comparison of the results

- Relatively small data
  - Validation set accuracy is higher than train set accuracy over the iterations

- Relatively big data
  - For almost every iteration, train set accuracy is higher than validation set accuracy
Other experiment results
Outcomes

Hyperparameter tuning may lead into overfitting on validation set; depends on:
- size of the data
- classifier
- hyperparameters of the classifier

Possible mitigation strategies:
- acquiring more data
- cross-validation
  - train-validation-test
Lessons learnt

- The importance of the data quality
- Complexity of computational time
  - Classifier
  - Data
- The importance of hyperparameters
Thanks!

Do you have any question?
References

1. Benner, J., Cross-Validation and Hyperparameter Tuning: How to Optimise your Machine Learning Model, Aug 6, 2020
   https://towardsdatascience.com/cross-validation-and-hyperparameter-tuning-how-to-optimise-your-machine-learning-model-13f005af9d7d

2. Bhadauriya, R., Cross-Validation(CV) and Hyper-Parameter Tuning, Sept 23
   https://medium.com/@cretrohit9/cross-validation-cv-and-hyper-parameter-tuning-5db4d209820d

3. Prashant Bhardwaj, Cross-Validation and Hyperparameter Tuning, Apr 8
   https://medium.com/almabetter/cross-validation-and-hyperparameter-tuning-91626c757428

4. Gorodetski, M., Hyperparameter Tuning Methods – Grid, Random or Bayesian Search?, Apr 28
   https://towardsdatascience.com/bayesian-optimization-for-hyperparameter-tuning-how-and-why-655b0ee0b399

5. Koehrsen, w., A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning, Jun 24, 2018

6. Dewancker, I., McCourt, M., Clark, S., Bayesian Optimization Primer

7. Neural Network Hyperparameter Tuning using Bayesian Optimization, 30/11/2021

   https://medium.com/@tzjy/strategy-for-deep-learning-hyper-parameter-tuning-bayesian-optimization-75a524a840d