Loss estimation

Blaž Erzar

In practice, many machine learning practitioners evaluate their models using train-test splits or cross-validation. When using these approaches, we are only estimating the true risk of the model. In this report, we present findings on how the estimated model risk compares to the true risk proxy in different scenarios.

A proxy for the true risk

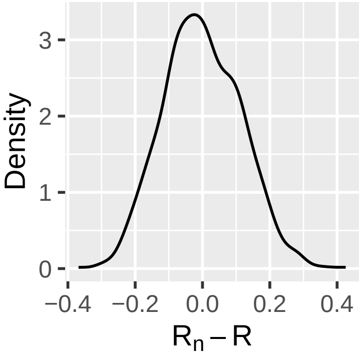
The model's risk converges to the true risk in probability. Because it is an average, it is distributed normally around the true value. Its variance drops with the sample size. To compute the variance of the log-loss, we train a model and compute the log-loss for every data instance. The standard error, i.e., the standard deviation of the estimator, is then the standard deviation of these log-losses divided by the square root of the sample size. We can see that the standard deviation of the log-losses is reduced to the third decimal digit.

```r
h_dgp <- glm(y ~ ., data = df_dgp, family = "binomial")
losses <- log_loss(df_dgp$y, predict(h_dgp, df_dgp, type = "response"))
cat(sprintf("Model's risk SE: %.4f\n", sd(losses) / sqrt(nrow(df_dgp))))
```

```
## Model's risk SE: 0.0017
```

Model loss estimator variability due to test data variability

First, we take a look at the variability of the model's risk computed on a separate test set, which is generated from the same DGP as the training data. In practice, we usually do not have access to such test sets.



```
## True risk proxy: 0.5807
## Mean difference: -0.0010
## 0.5-0.5 baseline true risk: 0.6931
## Median standard error: 0.1124
## Percentage of 95CI that contain the true risk proxy: 91.5%
```

Since the baseline true risk is higher than the model's true risk, the model successfully learned patterns from the data.

As expected by the convergence of the model's risk to the true risk, the difference is distributed with a mean close to zero. This means that the risk estimate using a test set is unbiased and a sensible estimate in practice if we have access to a test set from our DGP.

It can be seen from the standard error, that there is some variability in the estimate of the risk. This is expected because a smaller test set with 50 instances will not have the same distribution as the DGP. Since the estimate has some uncertainty, it should be reported in practice.

We also observe that the true risk is contained less than 95% of the time, meaning the standard error is slightly underestimated.

With a bigger train set, the true risk, as well as the estimate, would probably decrease, because the model would have more data to learn from. A bigger train set would probably also decrease the variance of the estimate because all such sets would be more representative of the DGP. If the train set was too small, the model would not be able to learn anything and could perform even worse than the baseline.

With a bigger test set, the variance of the difference would be smaller, because the test set would be more representative of the DGP. Similarly, for a smaller test set, the variance would be bigger.

Overestimation of the deployed model's risk

In practice, we usually deploy models that are trained on a dataset that is larger than the train set used to estimate the model's performance. Here, we compare the true risk of two models, h_1 trained on 50 observations and h_2 trained on an additional 50 observations. Both train sets have 50 observations in common.

```
## Summary of true risk h1 - true risk h2:
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.11351  0.05066  0.10773  0.31009  0.23238  6.01874
```

As expected, the model trained on a larger dataset performs better in general. The risk of h_2 is usually lower than the risk of h_1, meaning their difference is positive.

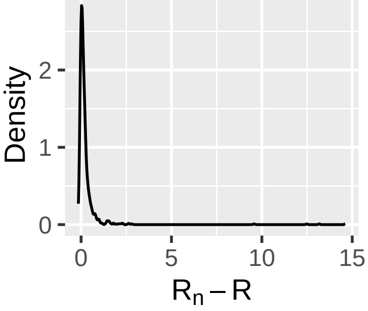
In practice, we should keep in mind that the risk estimation is computed on a different model than the one that is deployed. After deployment, we could additionally evaluate the model on a separate test set, perhaps even in parallel with the already deployed model, if it exists.

If the train sets were smaller, the difference would probably be even bigger, because each additional observation would have a larger impact. This is especially true because the models might not converge with such a small train set.

If the train sets were bigger, the difference would be smaller, because the smaller train set for h_1 would already be representative of the DGP and additional observations would not have a big impact.

Loss estimator variability due to split variability

Most commonly, we do not have access to an additional test set from the DGP, so we split the data into train and test sets.



```
## True risk proxy: 0.5266
## Mean difference: 0.2417
## Median standard error: 0.1226
## Percentage of 95CI that contain the true risk proxy: 85.9%
```

As expected, this estimate is highly biased. More accurately, it is positively biased and pessimistic. This means that the estimated risk is higher than the true risk. Also, the standard error is slightly higher than the standard error of separate test set estimates, but this error is even more underestimated. This can be seen from the coverage of the 95% confidence interval, which is much lower here. In each split, we get a variability of the model performance due to different train sets and a variability of the estimate due to different test sets.

There are also instances, where the estimated risk is much higher than the true risk. This is visible from the long right tail of the density plot. This probably occurs when the train set is not representative of the DGP, and the model cannot properly learn and performs much worse.

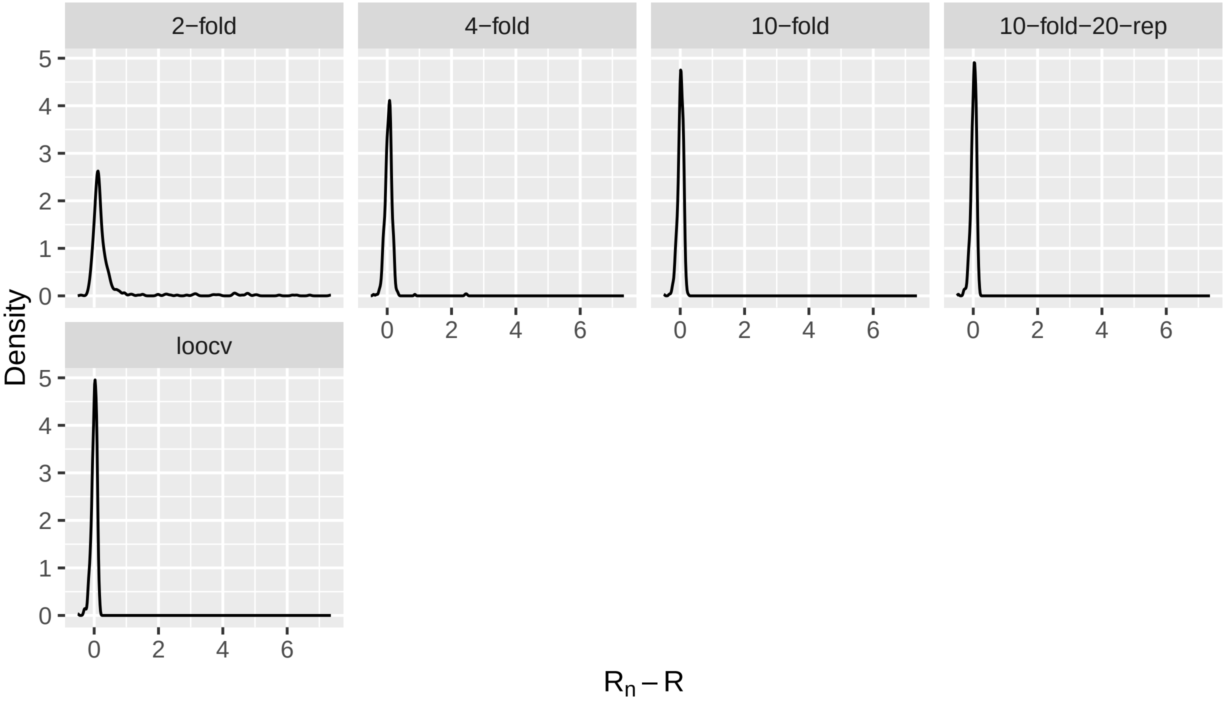
For practical purposes, we should note that train-test split estimation is highly biased, especially for small datasets, where a smaller train set causes a large decrease in the model's performance. Even if we report the uncertainty, it will usually be underestimated.

If the dataset was larger, the bias would probably be smaller, because the train set would be more representative of the DGP and the difference with the original train set would be smaller. The estimate would still have some variance, but it would be smaller if the dataset was large enough for its splits to be representative. Both the estimated and the true risk would be lower because the model would have more data to learn from.

If the proportion of the train set was bigger, the model would perform better and more similarly to the model trained on the whole dataset. This means that the bias would be smaller, but because the test set would be smaller, the variance would increase. On the other hand, a smaller proportion of the train set would increase the bias, but decrease the variance.

Cross-validation

Instead of train-test split, cross-validation is used many times. We compare cross-validation with a different number of folds and repetitions, while also drawing a new dataset from the DGP in each iteration.



We can observe that increasing the number of folds decreases the bias of the estimate. This is expected because the train set is getting bigger and more similar to the original dataset. This also means that the variance gets smaller with the number of folds because the model performance is more stable. However, drawing a new dataset in each iteration has caused the uncertainty to be more underestimated when compared to the previous task.

The smallest bias is achieved with the leave-one-out cross-validation, while 10-fold, 10-fold with 20 repetitions and LOO have similar variances. Out of these, 10-fold cross-validation with 20 repetitions underestimated the uncertainty the least.

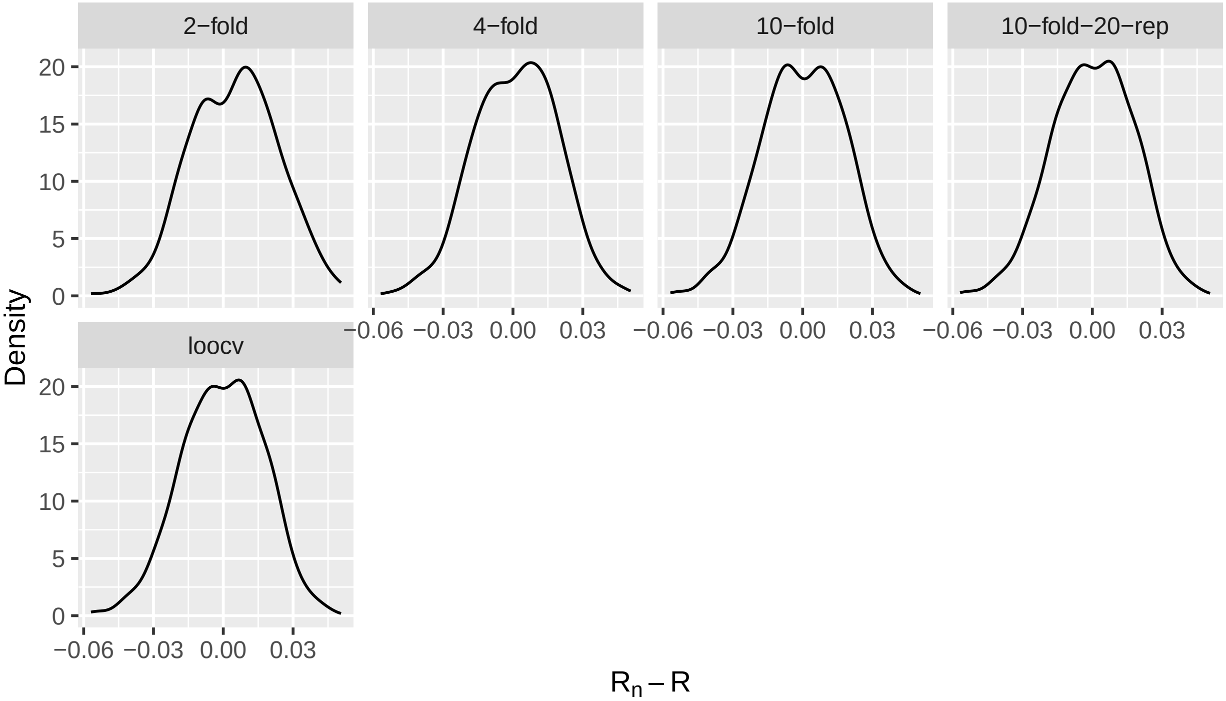
From these results, we can conclude that using 2-fold cross-validation is not the best option in practice. Usually, LOO performs the best, but we can achieve similar results with 10-fold cross-validation with 20 repetitions and save a lot of computational time.

```
## 2-fold:
##   Mean difference: 0.4434
##   Median standard error: 0.1113
##   Percentage of 95CI that contain the true risk proxy: 67.6%
## 4-fold:
##   Mean difference: 0.0490
##   Median standard error: 0.0829
##   Percentage of 95CI that contain the true risk proxy: 89.2%
## 10-fold:
##   Mean difference: 0.0086
##   Median standard error: 0.0769
##   Percentage of 95CI that contain the true risk proxy: 90.8%
## 10-fold-20-rep:
##   Mean difference: 0.0106
##   Median standard error: 0.0765
##   Percentage of 95CI that contain the true risk proxy: 92.8%
## loocv:
##   Mean difference: -0.0008
##   Median standard error: 0.0746
##   Percentage of 95CI that contain the true risk proxy: 91.8%
```

A different scenario

Increasing the size of the dataset from 100 to 1000, we can observe that the results are much different from the previous task. All methods perform very similarly, there being negligible differences between 10-fold, 10-fold with 20 repetitions and LOO. The 2-fold and 4-fold cross-validations perform only slightly worse, which might not even be significant.

The reason for this is the fact that the dataset is much bigger. Even the folds in the 2-fold cross-validation that had a high variance before, are now more representative of the DGP than the whole previous dataset was. We can also observe that this is the only task where the uncertainty is not underestimated, in some cases even being slightly overestimated.



```
## 2-fold:
##   Mean difference: 0.0048
##   Median standard error: 0.0179
##   Percentage of 95CI that contain the true risk proxy: 93.2%
## 4-fold:
##   Mean difference: 0.0012
##   Median standard error: 0.0176
##   Percentage of 95CI that contain the true risk proxy: 94.6%
## 10-fold:
##   Mean difference: 0.0003
##   Median standard error: 0.0175
##   Percentage of 95CI that contain the true risk proxy: 95.0%
## 10-fold-20-rep:
##   Mean difference: 0.0003
##   Median standard error: 0.0175
##   Percentage of 95CI that contain the true risk proxy: 95.4%
## loocv:
##   Mean difference: -0.0002
##   Median standard error: 0.0175
##   Percentage of 95CI that contain the true risk proxy: 95.6%
```