

# 13 Adaptive Resampling

Models can benefit significantly from tuning but the optimal values are rarely known beforehand.

`train` can be used to define a grid of possible points and resampling can be used to generate good estimates of performance for each tuning parameter combination. However, in the nominal resampling process, all the tuning parameter combinations are computed for all the resamples before a choice is made about which parameters are good and which are poor.

`caret` contains the ability to adaptively resample the tuning parameter grid in a way that concentrates on values that are the in the neighborhood of the optimal settings. See [this paper](#) for the details.

To illustrate, we will use the Sonar data from one of the [previous pages](#).

chemical mutagenicity data from [Kazius et al \(2005\)](#):

```
library(mlbench)
data(Sonar)
library(caret)
set.seed(998)
inTraining <- createDataPartition(Sonar$Class, p = .75, list = FALSE)
training <- Sonar[ inTraining,]
testing  <- Sonar[-inTraining,]
```

We will tune a support vector machine model using the same tuning strategy as before but with [random search](#):

```

svmControl <- trainControl(method = "repeatedcv",
                           number = 10, repeats = 10,
                           classProbs = TRUE,
                           summaryFunction = twoClassSummary,
                           search = "random")

set.seed(825)

svmFit <- train(Class ~ ., data = training,
               method = "svmRadial",
               trControl = svmControl,
               preProc = c("center", "scale"),
               metric = "ROC",
               tuneLength = 15)

```

Using this method, the optimal tuning parameters were a RBF kernel parameter of 0.0205 and a cost value of 44.8174899. To use the adaptive procedure, the `trainControl` option needs some additional arguments:

- `min` is the minimum number of resamples that will be used for each tuning parameter. The default value is 5 and increasing it will decrease the speed-up generated by adaptive resampling but should also increase the likelihood of finding a good model.
- `alpha` is a confidence level that is used to remove parameter settings. To date, this value has not shown much of an effect.
- `method` is either `"gls"` for a linear model or `"BT"` for a Bradley-Terry model. The latter may be more useful when you expect the model to do very well (e.g. an area under the ROC curve near 1) or when there are a large number of tuning parameter settings.
- `complete` is a logical value that specifies whether `train` should generate the full resampling set if it finds an optimal solution before the end of resampling. If you want to know the optimal parameter settings and don't care much for the estimated performance value, a value of `FALSE` would be appropriate here.

The new code is below. Recall that setting the random number seed just prior to the model fit will ensure the same resamples as well as the same random grid.

```

adaptControl <- trainControl(method = "adaptive_cv",
                             number = 10, repeats = 10,
                             adaptive = list(min = 5, alpha = 0.05,
                                              method = "gls", complete = TRUE),
                             classProbs = TRUE,
                             summaryFunction = twoClassSummary,
                             search = "random")

set.seed(825)

svmAdapt <- train(Class ~ ., data = training,
                 method = "svmRadial",
                 trControl = adaptControl,
                 preProc = c("center", "scale"),
                 metric = "ROC",
                 tuneLength = 15)

```

The search finalized the tuning parameters on the 19th iteration of resampling and was 5.5-fold faster than the original analysis. Here, the optimal tuning parameters were a RBF kernel parameter of 0.0205 and a cost value of 44.8174899. These match the previous settings even though 1293 fewer models were fit to the data.

Remember that this methodology is experimental, so please send any [questions or bug reports](#) to the package maintainer.