# There Has to Be an Easier Way: A Simple Alternative for Parameter Tuning of Supervised Learning Methods

Jill Lundell[1]
[1]Utah State University, Logan, UT 84321

**Abstract**
Several R packages can tune supervised learning methods, but some packages are so comprehensive they are difficult to use. Others are easier to use, but will only tune one or two methods. This paper presents an alternative R package that uses an optimizer to remove much of the frustration with parameter tuning for gradient boosting machines, support vector machines, and adaboost.

**Key Words:** Statistical learning, supervised learning, support vector machines, boosted trees, optimization, R

## 1. Introduction

Many tools exist in R to tune supervised learning models. The package caret (Kuhn 2008) provides an extensive set of tools for tuning many different supervised learning algorithms. However, caret is so extensive it is difficult to figure out how to use it to tune. Caret is also slow. Other packages contain intuitive and fast methods for tuning, but they are specific to certain models. For example, *e1071* (Myer et al. 2017) contains a function called *tune.svm* that is able to quickly tune a support vector machine (SVMs) with good results. However, if you wish to try several different learning methods, you have to hunt to find good tuning functions for each of the methods. It is also difficult to determine which parameters should be tuned and identify a reasonable range of values for those parameters.

The R package *EZtune* was designed to address many of these issues. The package will tune SVMs, adaboost, and gradient boosting machines (GBMs). A genetic algorithm or a quasi-Newton optimizer is used to determine an optimal set of tuning parameters. The user does not need know which parameters to tune or have a knowledge of a reasonable range of values for the tuning parameters. All three methods can be tuned using one function with only a few arguments. *EZtune* is a simple way for someone who is new to learning methods to find a model with optimal accuracy.

### 1.1 Format of the Package

The package *EZtune* consists of only two functions. The first function is called *EZtune* and has the form:

```
EZtune(x, y, type, method = "ada", optimizer = "ga", cv = FALSE,
fold = 10)
```

Where,

x = matrix of dependent variables
y = numeric vector of 0’s and 1’s
type = type of response; currently only a binary response is accepted
method = “ada” for adaboost, “svm” for SVMs, and “gbm” for GBMs
optimizer = “ga” for genetic algorithm or “optim” for quasi-Newton method
cv = FALSE for using resubstitution to assess accuracy, TRUE for using
    cross-validation to assess accuracy
fold = The number of folds for n-fold cross-validation. This is ignored
    if cv=FALSE

The function produces the following output:

- Object$summary: matrix of selected parameters and the final accuracy. Each estimate parameter can be called independently
- Object$nu, Object$cost, etc.
- Object$accuracy: best accuracy obtained by optimizer
- Object$best.model: the best model generated using optimized parameters
    - Adaboost: model produced by package *ada*
    - SVM: model produced by *e1071*
    - GBM: model produced by *gbm*
- Object$loss: returns the loss for *ada*
- Object$kernal: returns kernal for *svm*

The second function is called *EZtune.cv*. It takes the object produced by *EZtune* and computes a cross-validated accuracy. This function is useful for obtaining a better measure of accuracy when resubstitution is used to select tuning parameters. It is also useful for obtaining an average measure of accuracy over many simulations.

### 1.2 Methods Used in Constructing EZtune

*EZtune* was constructed using several R packages. The packages *e1071* (Myer et al. 2017), *ada* (Culp et al. 2016), and *gbm* (Ridgeway 2017) were used to generate the SVM models and the boosted trees. Models produced by *EZtune* are in the form of the package from which they were generated. Thus, the resulting models can be used as if they are a model from the respective package.

Optimization is done using the functions *optim* from the *stats* R package (R Core Team 2017) and *ga* from the package *GA* (Scrucca 2013). The function *optim* is used with the argument BFGS to optimize using a quasi-Newton method that was published simultaneously in 1970 by Broyden, Fletcher, Goldfarb and Shanno. The function *ga* was used to implement the genetic algorithm. The defaults for *ga* are used to optimize. Optimization was done exclusively on accuracy.

The current version of the package can only handle data with a binary response. A version that can address multiple classes and a continuous response is in development.

## 2. Performance Assessment

### 2.1 Set Up

Computations using *EZtune* were done on five datasets to assess the performance of the package: abalone, echocardiogram, ionosphere, Titanic, and lichen. All five datasets have a binary response. The titanic and lichen datasets have a test dataset associated with them that allows for more comprehensive evaluation of the package.

Datasets were cleaned prior to analysis. Missing values were removed from the datasets as were any variables that have too many categories, are too messy, are a case identifier, or are alternative response variables. None of the variables were transformed. Models were run on the remaining variables and observations 'as is'.

Each possible combination of the following was done for each dataset:

- Adaboost, GBM, and SVM
- Optimization with 10-fold cross-validation
- Optimization with resubstitution
- Optimization with quasi-Newton optimizer
- Optimization with genetic algorithm

The accuracy achieved by *EZtune* is reported for each run. *EZtune.cv* was run 10 times for each model and each of the cross-validated accuracies were averaged to obtain a mean cross-validated accuracy. The test data for the Titanic and lichen datasets were used to further assess the accuracy of the tuned model. The following sections show the results for the five datasets. The method, estimated parameters, optimized accuracies, and cross-validated accuracies are reported for each test.

#### *2.1.1 Abalone Data*

The abalone dataset is part of the *AppliedPredictiveModeling* package (Kuhn 2014) in R. It consists of seven continuous dependent variables and a response variable with three categories: male, female, and infant. Observations with an "infant" response were removed prior to calculations. The data have 2835 observations and do not have any missing values. Tables 1-3 show the results of the calculations. The largest accuracies in each table are in boldface type.

The results show that the when accuracy is optimized using resubstitution the computed accuracy is much higher than when cross-validation is used to optimize. However, when the cross-validated accuracies are compared for resubstitution and cross-validation optimizers they are very close. The cross-validated accuracies from the two different optimizers are also similar. This indicates that with the abalone data it does not matter if tuning is done using resubstitution or cross-validation to optimize accuracy. It also indicates that the genetic algorithm and the quasi-Newton optimizer perform equally well in this case.

**Table 1**: Results of *EZtune* with adaboost on the abalone data

| Method | Number of Iterations | Nu | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|
| GA Resub | 93 | 1.652 | **0.765** | 0.534 |
| GA CV | 38 | 0.054 | 0.558 | **0.557** |
| Optim Resub | 97 | 0.809 | 0.675 | 0.541 |
| Optim CV | 100 | 0.563 | 0.544 | 0.546 |

**Table 2**: Results of *EZtune* with gradient boosting machines on the abalone data

| Method | Interaction Depth | Number of Trees | Shrinkage | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|---|
| GA Resub | 5 | 3760 | 0.779 | **1.000** | 0.525 |
| GA CV | 3 | 4610 | 0.004 | 0.551 | 0.550 |
| Optim Resub | 2 | 500 | 0.100 | 0.703 | 0.537 |
| Optim CV | 1 | 498 | 0.100 | 0.555 | **0.558** |

**Table 3**: Results of *EZtune* with support vector machines on the abalone data

| Method | Epsilon | Cost | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|
| GA Resub | 2.272 | 98.960 | **0.621** | **0.552** |
| GA CV | 1.486 | 40.239 | 0.560 | 0.549 |
| Optim Resub | 0.100 | 0.647 | 0.563 | 0.550 |
| Optim CV | 0.823 | 0.654 | 0.553 | 0.549 |

### *2.1.2 Echocardiogram Data*

The echocardiogram dataset was obtained from the University of California Irvine Machine Learning Repository (Lichman 2013). The dataset consists of 13 variables and 133 observations. One of the variables is the number of months that a patient survived and another is a binary response about general survival. The variable for months survived was not used for analysis. All other variables were used for analysis. Observations with missing values were removed from the data prior to analysis. One-hundred and seven observations remain after the missing values are removed. Tables 4-6 show the results of the calculations. The largest accuracies in each table are in boldface type.

The results show that the when accuracy is optimized using resubstitution the computed accuracy is much higher than when cross-validation is used to optimize. However, when the cross-validated accuracies are compared for resubstitution and for cross-validation optimizers they are very close except for the genetic algorithm with GBM and SVM. The cross-validated accuracies from the different optimizers are also similar. This indicates that with the echocardiogram data it matters if tuning is done using resubstitution or cross-validation to optimize accuracy. It also indicates that the genetic algorithm and the quasi-Newton optimizer perform similarly well in this case.

**Table 4:** Results of *EZtune* with adaboost on the echocardiogram data.

| Method | Number of Iterations | Nu | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|
| GA Resub | 79 | 1.499 | **1.000** | 0.637 |
| GA CV | 14 | 0.114 | 0.738 | **0.690** |
| Optim Resub | 101 | 1.078 | **1.000** | 0.635 |
| Optim CV | 100 | 0.500 | 0.645 | 0.654 |

**Table 5:** Results of *EZtune* with gradient boosting machines on the echocardiogram data.

| Method | Interaction Depth | Number of Trees | Shrinkage | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|---|
| GA Resub | 8 | 1843 | 0.745 | **1.000** | 0.639 |
| GA CV | 8 | 2973 | 1.803 | 0.720 | 0.504 |
| Optim Resub | 2 | 500 | 0.100 | **1.000** | **0.644** |
| Optim CV | 2 | 500 | 0.100 | 0.617 | 0.620 |

**Table 6:** Results of *EZtune* with support vector machines on the echocardiogram data.

| Method | Epsilon | Cost | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|
| GA Resub | 0.787 | 95.202 | **0.972** | 0.580 |
| GA CV | 1.623 | 0.358 | 0.673 | **0.680** |
| Optim Resub | 0.100 | 1.000 | 0.813 | 0.639 |
| Optim CV | 0.167 | 2.921 | 0.626 | 0.621 |

### *2.1.3 Ionosphere Data*

The ionosphere dataset was obtained from the University of California Irvine Machine Learning Repository (Lichman 2013). The dataset consists of 34 continuous variables, a binary response variable, and 351 observations. There are no missing values in the dataset. Tables 7-9 show the results of the calculations. The largest accuracies in each table are in boldface type.

The results show that the when accuracy is optimized using resubstitution the computed accuracy is much higher than when cross-validation is used to optimize. However, when the cross-validated accuracies are compared for resubstitution and for cross-validation optimizers they are very close. The cross-validated accuracies from the different optimizers are also similar. This indicates that with the ionosphere data it does not matter if tuning is done using resubstitution or cross-validation to optimize accuracy. It also indicates that the genetic algorithm and the quasi-Newton optimizer perform similarly well in this case.

**Table 7**: Results of *EZtune* with adaboost on the ionosphere data.

| Method | Number of Iterations | Nu | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|
| GA Resub | 61 | 1.322 | **1.000** | 0.921 |
| GA CV | 60 | 0.158 | 0.949 | **0.932** |
| Optim Resub | 100 | 0.500 | **1.000** | 0.928 |
| Optim CV | 100 | 0.500 | 0.929 | 0.928 |

**Table 8:** Results of *EZtune* with gradient boosting machines on the ionosphere data.

| Method | Interaction Depth | Number of Trees | Shrinkage | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|---|
| GA Resub | 4 | 1617 | 0.559 | **1.000** | 0.893 |
| GA CV | 3 | 1189 | 0.141 | 0.937 | 0.930 |
| Optim Resub | 2 | 500 | 0.100 | **1.000** | 0.927 |
| Optim CV | 2 | 500 | 0.100 | 0.937 | **0.933** |

**Table 9:** Results of *EZtune* with support vector machines on the ionosphere data.

| Method | Epsilon | Cost | Accuracy | Cross-Validation Accuracy |
|---|---|---|---|---|
| GA Resub | 0.835 | 98.021 | **1.000** | 0.935 |
| GA CV | 0.707 | 19.879 | 0.957 | **0.951** |
| Optim Resub | 0.100 | 1.000 | 0.966 | 0.942 |
| Optim CV | 0.098 | 0.979 | 0.946 | 0.942 |

### *2.1.4 Titanic Data*

The Titanic dataset was obtained from the Kaggle website (www.kaggle.com). The dataset consists of 11 continuous and discrete predictor variables, a binary response variable, and 891 observations. The data contain several missing values. Four variables were removed because they are unique identifiers for the passengers. Missing values were imputed with the median for continuous variable or the most common factor for categorical variables. The Titanic dataset has a test dataset that can be used for additional verification of the models. The test dataset was prepared in the same manner as the trial dataset. Tables 10-12 show the results of the calculations including evaluation of the test dataset with each of the models. The largest accuracies in each table are in boldface type.

The results show that the when accuracy is optimized using resubstitution the computed accuracy is much higher than when cross-validation is used to optimize. However, when the cross-validated accuracies are compared for resubstitution and for cross-validation optimizers they are very close. Accuracies obtained form the test dataset show the same similarity between cross-validation and resubstitution, except in the case of adaboost. The test data accuracies from the cross-validated adaboost model are notably higher than for the models that were optimized on resubstitution accuracies. The test data accuracies for the model using the genetic algorithm with adaboost are also higher than those obtained from the quasi-Newton optimizer. Otherwise, the cross-validated accuracies from the different optimizers are similar. This indicates that with the Titanic data it does not matter if tuning is done using resubstitution or cross-validation to optimize accuracy except for adaboost. It also indicates that the genetic algorithm may perform a little better than the quasi-Newton optimizer for adaboost with these data.

**Table 10**: Results of *EZtune* with adaboost on the Titanic data.

| Method | Number of Iterations | Nu | Accuracy | Cross-Validation Accuracy | Test Data Accuracy |
|---|---|---|---|---|---|
| GA Resub | 76 | 1.616 | **0.976** | 0.790 | 0.679 |
| GA CV | 41 | 0.285 | 0.842 | **0.826** | **0.766** |
| Optim Resub | 98 | 1.533 | 0.973 | 0.793 | 0.646 |
| Optim CV | 100 | 0.500 | 0.824 | 0.815 | 0.703 |

**Table 11:** Results of *EZtune* with gradient boosting machines on the Titanic data.

| Method | Interaction Depth | Number of Trees | Shrinkage | Accuracy | Cross-Validation Accuracy | Test Data Accuracy |
|---|---|---|---|---|---|---|
| GA Resub | 4 | 4007 | 0.287 | **0.980** | 0.790 | 0.670 |
| GA CV | 5 | 2436 | 0.017 | 0.827 | 0.825 | 0.722 |
| Optim Resub | 2 | 500 | 0.100 | 0.895 | 0.825 | 0.722 |
| Optim CV | 2 | 500 | 0.100 | 0.828 | **0.829** | **0.727** |

**Table 12:** Results of *EZtune* with support vector machines on the Titanic data.

| Method | Epsilon | Cost | Accuracy | Cross-Validation Accuracy | Test Data Accuracy |
|---|---|---|---|---|---|
| GA Resub | 2.443 | 98.437 | **0.869** | 0.802 | 0.770 |
| GA CV | 0.319 | 8.587 | 0.829 | 0.820 | 0.780 |
| Optim Resub | 0.100 | 1.000 | 0.834 | **0.825** | **0.794** |
| Optim CV | 0.100 | 1.000 | 0.824 | **0.825** | **0.794** |

### *2.1.5 Lichen Data*

The lichen dataset was obtained from Cutler et al. (2007). The dataset consists of 53 continuous and discrete variables and 840 observations. There are no missing values in the dataset. The response of seven lichens are recorded in the dataset. The presences and absences of the lichen *Lobaria oregana* was used as the response variable for the calculations. The variables for the other six lichens were removed. The variable for plot number was also removed because it is a unique identifier. This dataset has a set of test data that were used to assess model accuracy. Tables 13-15 show the results of the calculations including evaluation of the test dataset with each of the models. The largest accuracies in each table are in boldface type.

The results show that the when accuracy is optimized using resubstitution the computed accuracy is much higher than when cross-validation is used to optimize. However, when the cross-validated accuracies are compared for resubstitution and for cross-validation optimizers they are very close to each other as they are for the other datasets. Accuracies obtained from the test dataset show the similar accuracies between cross-validation and resubstitution and between the two optimization methods. This indicates that with the lichen data it does not matter if tuning is done using resubstitution or cross-validation to optimize accuracy. It also indicates that the genetic algorithm and the quasi-Newton optimizer perform similarly well.

**Table 13**: Results of *EZtune* with adaboost on the lichen data.

| Method | Number of Iterations | Nu | Accuracy | Cross-Validation Accuracy | Test Data Accuracy |
|---|---|---|---|---|---|
| GA Resub | 45 | 1.085 | **1.000** | 0.831 | 0.614 |
| GA CV | 74 | 0.701 | 0.863 | 0.838 | **0.636** |
| Optim Resub | 100 | 0.500 | **1.000** | 0.843 | 0.630 |
| Optim CV | 102 | 0.328 | 0.843 | **0.846** | **0.636** |

**Table 14:** Results of *EZtune* with gradient boosting machines on the lichen data.

| Method | Interaction Depth | Number of Trees | Shrinkage | Accuracy | Cross-Validation Accuracy | Test Data Accuracy |
|---|---|---|---|---|---|---|
| GA Resub | 6 | 1926 | 0.503 | **1.000** | 0.831 | 0.618 |
| GA CV | 3 | 1104 | 0.019 | 0.851 | **0.845** | **0.629** |
| Optim Resub | 3 | 501 | 0.197 | **1.000** | 0.834 | 0.623 |
| Optim CV | 2 | 500 | 0.100 | 0.829 | 0.837 | 0.613 |

**Table 15:** Results of *EZtune* with support vector machines on the lichen data.

| Method | Epsilon | Cost | Accuracy | Cross-Validation Accuracy | Test Data Accuracy |
|---|---|---|---|---|---|
| GA Resub | 2.567 | 98.132 | **0.995** | 0.829 | 0.611 |
| GA CV | 1.699 | 5.524 | 0.846 | 0.840 | 0.619 |
| Optim Resub | 0.100 | 1.000 | 0.887 | 0.845 | **0.650** |
| Optim CV | 0.086 | 1.470 | 0.848 | **0.849** | 0.638 |

### *2.1.5 Summary of Results*

All five of the datasets show similar results. The optimized accuracies obtained from *EZtune* are higher when resubstitution is used for optimization rather than cross-validation. However, when a cross-validation accuracy is obtained from the model tuned using resubstitution, the accuracy is very close to that of the model tuned using cross-validation with only one exception. Tuning is much faster when resubstitution is used to optimize than when cross-validation is used to optimize. The results from the 5 datasets indicate that it may be sufficient to tune the model using resubstitution and then use *EZtune.cv* to obtain a better accuracy. The results also indicate that the quasi-Newton optimizer and the genetic algorithm work similarly well in most cases.

## 3. Future Work

The package *EZtune* is currently working well for binary datasets. It can be downloaded from https://github.com/jillbo1000/EZtune. Future developments include:

- Options for continuous response
- Options for a response with more than two classes
- Upgrade code for greater speed and optimization
- Investigate use of other optimizers
- Add features for handling large datasets in a reasonable amount of time
- Add simple ways to alter additional parameters such as kernal and loss
- Incorporate other packages
- Add an option to optimize on the area under the ROC curve
- Make a vignette
- Post the package on CRAN

## Acknowledgements

I would like to thank my advisor, Dr. Richard Cutler, for his advice and support on this project.

## References

Culp, Mark, Kjell Johnson and George Michailidis (2016). *ada: The R Package Ada for Stochastic Boosting*. https://CRAN.R-project.org/package=ada.

Cutler, D. Richard., Thomas C. Edwards Jr., Karen H. Beard, Adele Cutler, Kyle T. Hess, Jacob Gibson, and Joshua J. Lawler. 2007. Random Forests for Classification in Ecology. *Ecology* **88**(11): 2783—2792.

Kuhn, Max. (2008). "Caret package." *Journal of Statistical Software*, 28(5)

Kuhn, Max and Kjell Johnson (2014). *AppliedPredictiveModeling: Functions and Data Sets for 'Applied Predictive Modeling*. https://CRAN.R-project.org/package=AppliedPredictiveModeling.

Lichman, Moshe. (2013). *UCI Machine Learning Repository*, Irvine, CA: University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml.

Meyer, David, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2017). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group*, https://CRAN.R-project.org/package=e1071.

R Core Team (2017). R: *A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Ridgeway, Greg (2017). *gbm: Generalized Boosted Regression*. https://CRAN.R-project.org/package=gbm.

Scrucca, Luca (2013). "GA: A Package for Genetic Algorithms in R." *Journal of Statistical Software*, 53(4), 1-37.