SCIENTIFIC TEST
& ANALYSIS
TECHNIQUES
CENTER OF EXCELLENCE

Data-Driven Model Development

August 2022

Dr. Anthony Sgambellone, Contractor

The STAT COE provides independent STAT consultation to designated acquisition programs and special projects to improve Test & Evaluation (T&E) rigor, effectiveness, & efficiency.

About this Publication:
This work was conducted by the Scientific Test & Analysis Techniques Center of Excellence under contract FA8075-18-D-0002, Task FA8075-21-F-0074.

For more information:
Visit, www.AFIT.edu/STAT
Email, CoE@afit.edu
Call, 937-255-3636 x4736

Scientific Test & Analysis Techniques Center of Excellence
2950 Hobson Way
Wright-Patterson Air Force Base, Ohio

Version: 1, FY22

Modernizing *the* Culture *of* Test *&* Evaluation

Executive Summary

There are many techniques to choose from when training data-driven models to ensure that one finds the best fit; however, it is vital to avoid over-fitting a model. Over-fitting occurs when a model “learns” to fit the training set so well that it begins to fit the random sampling variation as if it were predictive. To address this challenge, this Best Practice reviews fitting principles such as examining multiple models and decision metrics and offers model developers methods to avoid over-fitting models. We present methods including Train/Validate/Test and Cross-Validation techniques, as well as fundamental principles such as never to validate the model on training data. For this Best Practice it is assumed that the user has already formulated a well-defined purpose for the model and has appropriate training data. This paper aims to help model developers select appropriate training and validation methods to best isolate true predictive relationships from the noise of a system.

Keywords: Model, Modeling, Training, Validation

Table of Contents

Introduction

Data is well recognized as a source of insight in a wide variety of fields. However, reliable insights typically require more than examination of the raw values. Data-driven mathematical models are valuable tools for prediction or characterization of a system. With modern software they are also deceptively easy to implement, but care must be taken if the results are to be useful. Machine Learning models, despite their name, have no awareness of the concepts that we use them to describe.

If a model is trained carelessly it will build 'superstitions,' using random variation as if it were predictive. Although it is not possible to eliminate that risk, techniques do exist to minimize that effect and then measure the reliability of a model for future events. This Best Practice introduces these techniques so that users avoid missteps that may result in severely misleading models. More specifically, it will cover three prominent issues in training a model to data: over-fitting, generalizability, and model comparison. This paper will also describe standard techniques such as training versus validation sets, cross-validation, and out of sample validation to potentially implement with said principles for any algorithm/model that may fit the reader's needs.

Observational Data Limits

Data-driven model training is frequently performed where the system is too complex to model (exclusively) through known causal relations. Observational data is commonly used because it is typically easier to gather and is effective for prediction. Further measures would be necessary to imply causation, but some of the techniques below include measures to examine the resilience of predictive relationships.

The methods below assume that the modeler already has a well-defined purpose for the model tied to a practical, measurable response variable, as well as having already collected appropriate training data.

Validation Principles

Here we present general principles that should *always* apply to any method when training a data-driven model, before we discuss specific applications of these principles. The following principles are:

- always validate,
- do not validate on training data,
- use multiple models and methods, and
- examine multiple decision metrics.

Always Validate

Sampling variation or inherent noise in a system can not only mask important relationships, but it can also create false appearance of relationships that would not hold for future observations. The more complex and precise a model is made, the more likely it is to pick up false patterns of random variation as if it were predictive. Validation on "new" data is a safeguard against such over-fitting, preserving the usefulness of a model.

Do Not Validate on Training Data

Do not validate on data that was used to train the model. It is already known that training data appear to have any relationship that the model developed, whether it is a true relationship or

chance. Validating on data used to train the model, or adjusting the model from the validation data, reports a self-fulfilling prophecy and provides none of the safeguards associated with validation.

Use Multiple Models and Methods

Data-driven model training is an exploratory process. Build multiple models for comparison and use multiple types of models when possible and acceptable for your model's purpose.

Examine Multiple Decision Metrics

For a given type of response there will be several metrics by which to measure goodness of fit and predictive value, and each will report on different aspects of the fit. The most common metrics measure an overall fit throughout the range of the model. Such metrics should be examined but may not be the most appropriate for a given model's purpose.

For example, common metrics to measure predictive power for a quantitative response include Adjusted R^2 and AIC. These are robust metrics to measure total fit, but if a model is designed to estimate the potential benefit of an opportunity, then a more targeted metric could better suit the purpose. For example, it may be of particular interest not to over-estimate the benefit of opportunities near the threshold level for a decision.

Also, graphical metrics that display goodness of fit or predictive value over the range of predictors or depth of a population provide much more information for a robust decision than single-number summaries.

Each of the principles above should be applied in any validation. There are many methods in which this can be done; each with its own requirements, strengths, and weaknesses. The next section describes how to implement a few of these methods.

Validation Strategies

It is important to validate a trained model on new data. If training data is used to assess the performance of a model, then the result is likely to overestimate the value of the model. A model will often pick up spurious relationships, things that only seemed related due to chance and will not hold in future observations. It is impossible to find or guard against spurious relationships with the data that was used to create them.

The validation method should be chosen before model training begins. The validation structure should be chosen and implemented before dimension reduction for parameter selection, if applicable. The best validation method to choose depends upon the data available and the purpose of the model. We will describe two methods below: Train/Validation/Test and Cross-Validation.

The Train/Validate/Test Validation Method

A comprehensive validation method is to divide the available data into three sets: one for model training, another for model comparison and selection, and one to test performance. A common format is to use 50%-80% of the data for training and split the remainder for the validation and test sets. The split should be performed by random selection so that each set remains as representative of the population as possible.

This three-way split is the preferred method when there is plenty of data. There must be enough data in the training set to properly identify relationships, with enough left over to provide a good

estimate of fit vs over-fit in the validation set, and to confirm performance in the test set.

The training set, and only the training set, should be used for attribute selection and model parameter estimation. The validation set is used to compare models. Since the validation set was randomly split from the training set, it should not maintain most spurious relationships from the training set. A large difference in performance between the training and validation sets is an indicator of having over-fit the model.

If the validation set guards against over-fitting, then why do we need the test set? Some models will include spurious relations in the training that overlap with some spurious relations in the validation set by chance. This is typically minimal but can be exaggerated by fitting many models and choosing the one that maximizes apparent performance on the validation set. In addition, many model algorithms use the validation set to determine when to stop fitting. Any model decision to improve performance that is made using the validation set is a potential source of over-fitting. Hence the need for a final test set that was not used in any stage of model design. The model performance on the test set should be similar to that of the validation set: a large drop in performance indicates over-fitting.

The Use of Out of Sample or Out of Time Sets

Alternative approaches to the test set are known as Out Of Sample (OOS) or Out Of Time (OOT). In these cases, the original data is split between only the training and validation sets. For OOS the test set data is taken from a different population that is expected to be like the one of primary interest: this could be data for a similar device or with participants from a different region, etc. An OOT sample would be taken from the same population but at a significantly removed time: typically, older data is used but it could also be from data that had not yet been collected when the modeling project began.

The purpose of both OOS and OOT are to provide insight into the persistence of relationships. This is particularly important for observational studies and for models focused on prediction rather than explanation. Model terms may not have a causal relationship to the response: there may be a lurking causal factor which happens to be correlated with the model attribute, a relationship that may or may not persist. Continued good performance on an OOS or OOT sample is support for the reliability of predictive relationships used by the model.

Model performance on OOS and/or OOT sets is not expected to be as powerful as on the validation set, but an unexplained large drop in performance would be cause for concern for the life expectancy or generalizability of the model, in addition to checking for over-fit.

Cross-Validation Methods

Cross-Validation is an approach used when there is not enough data for a useful Training/Validation/Test split. The entire dataset is used to train the model. It is not possible to create a direct comparison to estimate the over-fitting of the model. However, the data is then split into “Training” and “Validation” sets, and the same method, with the same attribute selection and stopping rules, is used to build a model on the training set, and the fit of the model trained on this reduced set is measured on the validation set. This model trained on the reduced training set will be different from the model designed on the total data set.

This process is then repeated several times with different splits for training and validation. The model is trained the same way on each training set, which will result in somewhat different models, and each model’s performance is measured on the corresponding validation set. The performance metrics for all the training/validation splits are averaged for the final fit metrics. This

will not give a direct measure of the over-fit of the total model, but it does provide a measure of the degree to which that method tends to over-fit relationships for this population and sample. This can be repeated for each candidate model for model selection.

There are several approaches to cross-validation, and a few of them are described in greater detail below.

Leave-p-Out (LpO)

Choose a small value, say p=2. Fit the model on all but two observations and evaluate the model on the remaining two observations. Repeat for all possible splits for this value of *p*. LpO is often recommended in the context of a binary response.

K-Fold

Divide the data into *k* equal sets, say 10. Fit the model on all but one of the sets, evaluating fit on the remaining set that was left out. Repeat leaving a different set out of the training until each set has been left out once.

Stratified

Rather than a type of cross-validation, stratified sampling is a technique that can be applied in the creation of splits in any cross-validation method. When a particular subset of interest in the population is rare, it is possible that random splits will not result in an appropriately representative proportion. To account for this, first divide the dataset into "strata" based on the characteristic(s) of interest. Perform the training/validation split separately for each stratum and then recombine for one training and one validation set. This technique may also apply Train/Validate/Test splits but is more likely to be important in the smaller sample sizes that motivate the use of cross-validation.

We have briefly covered a variety of strategies that would meet the needs for most training and validation of data-driven models. Next we walk through an example that demonstrates some issues that might arise in practice.

Illustrative Example

Consider the modeling of a continuous response variable with sufficient data for a Train/Validate/Test split. Three models have been fit using the training data set: a decision tree, forward stepwise regression, and backwards stepwise regression. Fit statistics are shown in Figure 1.

Decision Tree		
R^2	RMSE	AICc
0.876	0.158	502.375

Forwards Regression		
R^2	RMSE	AICc
0.644	0.308	398.518

Backwards Regression		
R^2	RMSE	AICc
0.721	0.293	410.372

Figure 1

Model Performance on the Training Set

At initial glance the decision tree seems to be the most powerful model, explaining the most variation from the system. Lower AICc is desired though and is a more robust measure of model fit. The regression models have very similar lower AICc, with the backwards stepwise regression capturing more variation of the system according to the higher explained variance R^2 and a correspondingly lower Root Mean Squared Error (RMSE). However, the model's performance on the validation set, shown in Figure 2, is of greater importance for model

comparison and projected usefulness.

Decision Tree		
R^2	RMSE	AICc
0.52	0.368	660.518

Forwards Regression		
R^2	RMSE	AICc
0.619	0.329	401.351

Backwards Regression		
R^2	RMSE	AICc
0.621	0.328	399.853

Figure 2
Model Performance on the Validation Set

The decision tree has a comparatively poor fit on the validation data in this case. It has lower R2 and higher RMSE and AICc than either regression model. Further, the decision tree metrics have a much greater difference between the training and validation sets than these two regression models. This would indicate a troubling degree of over-fit on the training data.

The regression models in this case provide a much better prediction on the validation data. The backwards stepwise regression seems to perform slightly better than the forwards stepwise regression, but the differences are very small and likely due to sampling variation. Although the difference from training to validation for backwards stepwise regression is not extreme, the forwards stepwise regression model has very similar performance on the validation and almost identical degree of prediction to its performance on the training set, showing no sign of over-fit. Thus we will choose the forward stepwise regression model in this case. Finally, we check this model's performance on the test set.

The test set also has similar performance which indicates that there was no over-fit on the validation data, for example from excessive model comparison and refinement on the validation set. This model is likely to perform similarly well on future observations.

Conclusion

Data-driven models have extraordinary potential to predict performance in systems too complex to sufficiently model through causal relations, but care must be taken if the resulting model is to be useful. The modeling process is frequently a significant undertaking between infrastructure to hold and manipulate data, collection of the data itself, modeling, and implementation. If validation principles are ignored or misused, the resulting model may have disastrously poor performance in operation which can cause losses far beyond the resources spent to create it.

The principles and techniques outlined in this paper should provide a good foundation for training data-driven models and obtaining reliable expectations of performance. When attempting to find the best fit use multiple metrics and metrics tailored to your use case, and use validation techniques such as those above that do not validate a model on the data that was used to train it.