Test set

From Wikipedia, the free encyclopedia

In many areas of information science, finding predictive relationships from data is a very important task. Initial discovery of relationships is usually done with a *training set* while a *test set* and *validation set* are used for evaluating whether the discovered relationships hold. More formally, a **training set** is a set of data used to discover potentially predictive relationships. A **test set** is a set of data used to assess the strength and utility of a predictive relationship. Test and training sets are used in intelligent systems, machine learning, genetic programming and statistics.

Contents

Rationale

Regression analysis was one of the earliest such approaches to be developed. The data used to construct or discover a predictive relationship are called the training data set. Most approaches that search through training data for empirical relationships tend to overfit the data, meaning that they can identify apparent relationships in the training data that do not hold in general. A test set is a set of data that is independent of the training data, but that follows the same probability distribution as the training data. If a model fit to the training set also fits the test set well, minimal overfitting has taken place. A better fitting of the training set as opposed to the test set usually points to overfitting.

Validation set

In order to avoid overfitting, when any classification parameter needs to be adjusted, it is necessary to have a **validation set** in addition to the training and test sets. For example if the most suitable classifier for the problem is sought, the training set is used to train the candidate algorithms, the validation set is used to compare their performances and decide which one to take, and finally, the test set is used to obtain the performance characteristics such as accuracy, sensitivity, specificity, F-measure and so on. The validation set functions as a hybrid: it is training data used by testing, but neither as part of the low-level training, nor as part of the final testing.

Most simply, part of the training set can be set aside and used as a validation set; this is known as the **holdout method**, and common proportions are 70%/30% training/validation. Alternatively, this process can be repeated, repeatedly partitioning the original training set into a training set and a validation set; this is known as cross-validation. These repeated partitions can be done in various ways, such as dividing into 2 equal sets and using them as training/validation and then validation/training, or repeatedly selecting a random subset as a validation set.

These can be defined as:[1][2]

- Training set: A set of examples used for learning, that is to fit the parameters [i.e., weights] of the classifier.
- Validation set: A set of examples used to tune the hyperparameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.
- Test set: A set of examples used only to assess the performance [generalization] of a fully-specified classifier.

The basic process of using a validation set for model selection (as part of training set, validation set, and test set) is:[3][2]

> Since our goal is to find the network having the best performance on new data, the simplest approach to the comparison of different networks is to evaluate the error function using data which is independent of that used for training. Various networks are trained by minimization of an appropriate error function defined with respect to a training data set. The performance of the networks is then compared by evaluating the error function using an independent validation set, and the network having the smallest error with respect to the validation set is selected. This approach is called the hold out method. Since this procedure can itself lead to some overfitting to the validation set, the performance of the selected network should be confirmed by measuring its performance on a third independent set of data called a test set.

An application of this process is in early stopping, where the candidate models are successive iterations of the same network, and training stops when the error on the validation set grows, choosing the previous model (the one with minimum error).

Sometimes the training set and validation set are referred to collectively as **design set**: the first part of the design set is the training set, the second part is the validation step.[4]

Hierarchical classification

Another example of parameter adjustment is **hierarchical classification** (sometimes referred to as **instance space decomposition** [5]), which splits a complete multi-class problem into a set of smaller classification problems. It serves for learning more accurate concepts due to simpler classification boundaries in subtasks and individual feature selection procedures for subtasks. When doing classification decomposition, the central choice is the order of combination of smaller classification steps, called the classification path. Depending on the application, it can be derived from the confusion matrix and, uncovering the reasons for typical errors and finding ways to prevent the system make those in the future. For example,[6] on the validation set one can see which classes are most frequently mutually confused by the system and then the instance space decomposition is done as follows: firstly, the classification is done among well recognizable classes, and the difficult to separate classes are treated as a single joint class, and finally, as a second classification step the joint class is classified into the two initially mutually confused classes.

Use in artificial intelligence, machine learning, and statistics

In artificial intelligence or machine learning, a training set consists of an input vector and an *answer* vector, and is used together with a supervised learning method to *train* a knowledge database (e.g. a neural net or a naive Bayes classifier) used by an AI machine. Validation sets can be used for regularization by early stopping: stop training when the error on the validation set increases, as this is a sign of overfitting to the training set.[7]

This simple procedure is complicated in practice by the fact that the validation error may fluctuate during training, producing multiple local minima. This complication has led to the creation of many ad-hoc rules for deciding when overfitting has truly begun.[7]

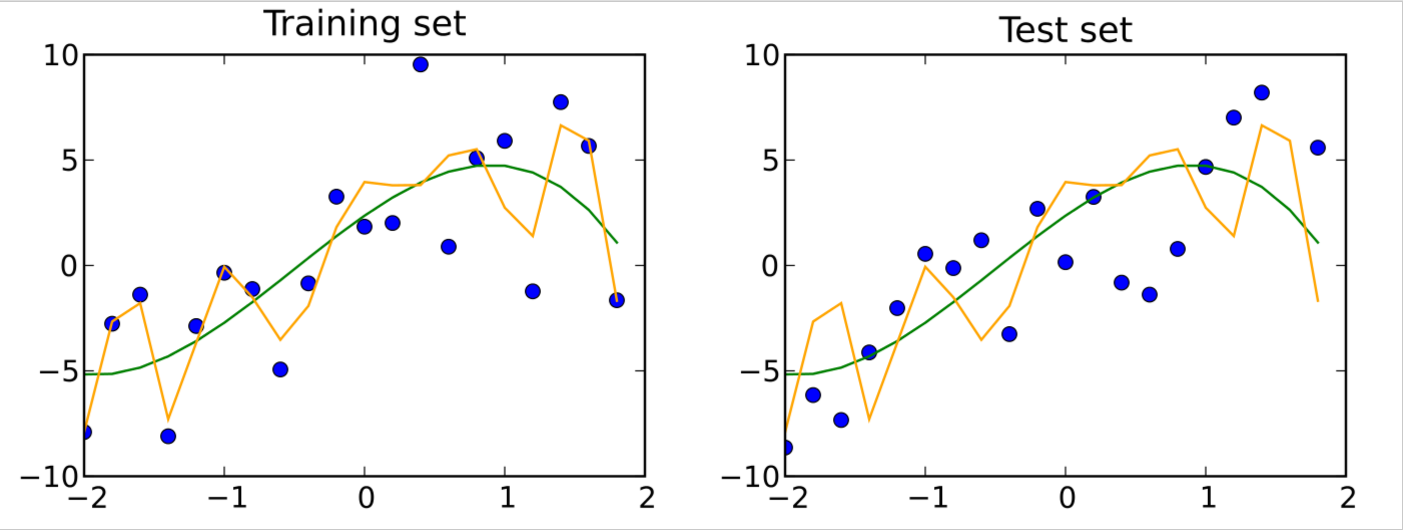
In statistical modeling, a training set is used to fit a model that can be used to predict a "response value" from one or more "predictors." The fitting can include both variable selection and parameter estimation. Statistical models used for prediction are often called regression models, of which linear regression and logistic regression are two examples.

In these fields, a major emphasis is placed on avoiding overfitting, so as to achieve the best possible performance on an independent **test set** that follows the same probability distribution as the training set.

Use in intelligent systems

In general, an intelligent system consists of a function taking one or more arguments and results in an output vector, and the learning method's task is to run the system once with the input vector as the arguments, calculating the output vector, comparing it with the answer vector and then changing somewhat in order to get an output vector more like the answer vector next time the system is simulated.

Example



A training set (left) and a test set (right) from the same statistical population are shown as blue points. Two predictive models are fit to the training data. Both fitted models are plotted with both the training and test sets. In the training set, the MSE of the fit shown in orange is 4 whereas the MSE for the fit shown in green is 9. In the test set, the MSE for the fit shown in orange is 15 and the MSE for the fit shown in green is 13. The orange curve severely overfits the training data, since its MSE increases by almost a factor of four when comparing the test set to the training set. The green curve overfits the training data much less, as its MSE increases by less than a factor of 2.

See also

- Cross-validation (statistics)
- Machine learning
- Statistical Classification

References

1. Ripley, B.D. (1996) *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, p. 354
2. "Subject: What are the population, sample, training set, design set, validation set, and test set? (ftp://ftp.sas.com/pub/neural/FAQ.html#A_data)", Neural Network FAQ, part 1 of 7: Introduction (ftp://ftp.sas.com/pub/neural/FAQ.html) (txt (ftp://ftp.sas.com/pub/neural/FAQ1.txt)), comp.ai.neural-nets, Sarle, W.S., ed. (1997, last modified 2002-05-17)
3. Bishop, C.M. (1995), *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press, p. 372
4. *Statistical and Neural Classifiers: An Integrated Approach to Design*, by Sarunas Raudys (2012), p. 2 (https://books.google.com/books?id=W94LBwAAQBAJ&pg=PA2&dq=%22design+set%22), p. 212 (https://books.google.com/books?id=W94LBwAAQBAJ&pg=PA212&dq=%22design+set%22)

5. Cohen S, Rokach L., Maimon O. Decision-tree instance-space decomposition with grouped gain-ratio In J. Information Sciences, vol. 177, issue 17, pp. 3592–3612. Elsevier. 2007.
6. Sidorova, J., Badia, T. "ESEDA: tool for enhanced speech emotion detection and analysis". The 4th International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution (AXMEDIS 2008). Florence, November, 17-19, pp. 257–260. IEEE press.
7. Prechelt, Lutz; Geneviève B. Orr (2012-01-01). "Early Stopping — But When?". In Grégoire Montavon, Klaus-Robert Müller (eds.). *Neural Networks: Tricks of the Trade*. Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp. 53–67. ISBN 978-3-642-35289-8. Retrieved 2013-12-15.

External links

- Foundations of Genetic Programming (http://www.cs.ucl.ac.uk/staff/W.Langdon/FOGP/)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Test_set&oldid=726227103"

Categories: Datasets in machine learning | Machine learning | Data analysis