

Parametric and Nonparametric Machine Learning Algorithms

by [Jason Brownlee](#) on March 14, 2016 in [Understand Machine Learning Algorithms](#)

What is a [parametric](#) machine learning algorithm and how is it [different](#) from a [nonparametric](#) machine learning algorithm?

In this post you will discover the [difference](#) between [parametric](#) and [nonparametric](#) machine learning algorithms.

Learning a Function

Machine learning can be summarized as learning a function (f) that maps input variables (X) to output variables (Y).

$$Y = f(x)$$

An algorithm learns this target mapping function from [training](#) data.

The form of the function is unknown, so our job as machine learning practitioners is to evaluate different machine learning algorithms and see which is better at approximating the underlying function.

[Different algorithms](#) make different [assumptions](#) or [biases](#) about the form of the function and how it can be learned.

Parametric Machine Learning Algorithms

Assumptions can greatly simplify the learning process, but can also limit what can be learned. Algorithms that simplify the function to a known form are called parametric machine learning algorithms.

A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a [parametric](#) model. No matter how much data you throw at a [parametric](#) model, it won't change its mind about how many parameters it needs.

— [Artificial Intelligence: A Modern Approach](#), page 737

The algorithms involve **two steps**:

1. Select a form for the function.
2. Learn the **coefficients** for the function from the **training** data.

An easy to understand functional form for the mapping function is a line, as is used in linear regression:

$$b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 = 0$$

Where **b0**, **b1** and **b2** are the **coefficients** of the line that control the intercept and slope, and **x1** and **x2** are two input variables.

Assuming the functional form of a line greatly simplifies the learning process. Now, all we need to do is estimate the coefficients of the line equation and we have a predictive model for the problem.

Often the assumed functional form is a linear combination of the input variables and as such parametric machine learning algorithms are often also called “*linear machine learning algorithms*”.

The problem is, the actual unknown underlying function may not be a linear function like a line. It could be almost a line and require some minor transformation of the input data to work right. Or it could be nothing like a line in which case the assumption is wrong and the approach will produce poor results.

Some more examples of parametric machine learning algorithms include:

- Logistic Regression
- Linear Discriminant Analysis
- Perceptron
- Naive Bayes
- Simple Neural Networks

Benefits of Parametric Machine Learning Algorithms:

- **Simpler**: These methods are easier to understand and interpret results.
- **Speed**: Parametric models are very **fast to learn** from data.
- **Less Data**: They do not require as much training data and can work well even if the fit to the data is not perfect.

Limitations of Parametric Machine Learning Algorithms:

- **Constrained:** By choosing a functional form these methods are highly constrained to the specified form.
- **Limited Complexity:** The methods are more suited to simpler problems.
- **Poor Fit:** In practice the methods are unlikely to match the underlying mapping function.

Nonparametric Machine Learning Algorithms

Algorithms that do not make strong assumptions about the form of the mapping function are called nonparametric machine learning algorithms. By not making assumptions, they are free to learn any functional form from the training data.

Nonparametric methods are good when you have [a lot of data](#) and [no prior knowledge](#), and when you don't want to worry too much about [choosing just the right features](#).

— [Artificial Intelligence: A Modern Approach](#), page 757

[Nonparametric](#) methods seek to best fit the training data in constructing the mapping function, whilst maintaining some ability to [generalize](#) to unseen data. As such, they are able to fit a large number of functional forms.

An easy to understand [nonparametric](#) model is the k-nearest neighbors algorithm that makes predictions based on the **k** most similar training patterns for a new data instance. The method does not assume anything about the form of the mapping function other than patterns that are close are likely have a similar output variable.

Some more examples of popular nonparametric machine learning algorithms are:

- k-Nearest Neighbors
- Decision Trees like CART and C4.5
- Support Vector Machines

[Benefits](#) of Nonparametric Machine Learning Algorithms:

- **Flexibility:** Capable of fitting a large number of functional forms.
- **Power:** No assumptions (or weak assumptions) about the underlying function.
- **Performance:** Can result in higher performance models for prediction.

[Limitations](#) of Nonparametric Machine Learning Algorithms:

- **More data:** Require a lot more training data to estimate the mapping function.
- **Slower:** A lot slower to train as they often have far more parameters to train.

- **Overfitting:** More of a risk to overfit the training data and it is harder to explain why specific predictions are made.

Further Reading

This section lists some resources if you are looking to learn more about the difference between parametric and non-parametric machine learning algorithms.

Books

- [An Introduction to Statistical Learning: with Applications in R](#), chapter 2
- [Artificial Intelligence: A Modern Approach](#), chapter 18

Posts

- [What are the advantages of using non-parametric methods in machine learning?](#) on Quora
- [What are the disadvantages of non-parametric methods in machine learning?](#) on Quora
- [Nonparametric statistics](#) on Wikipedia
- [Parametric statistics](#) on Wikipedia
- [Parametric vs. Nonparametric](#) on Stack Exchange

Summary

In this post you have discovered the **difference** between **parametric** and **nonparametric** machine learning algorithms.

You learned that **parametric** methods make **large assumptions** about the mapping of the input variables to the output variable and in turn are faster to train, require less data but may not be as powerful.

You also learned that **nonparametric** methods make few or **no assumptions** about the target function and in turn require a lot more data, are **slower to train** and have a higher model complexity but can result in more powerful models.