# Support Vector Machines for Machine Learning

by **Jason Brownlee** on April 20, 2016 in **Machine Learning Algorithms**

Tweet Share Share

Last Updated on August 12, 2019

Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms.

They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with little tuning.

In this post you will discover the Support Vector Machine (SVM) machine learning algorithm. After reading this post you will know:

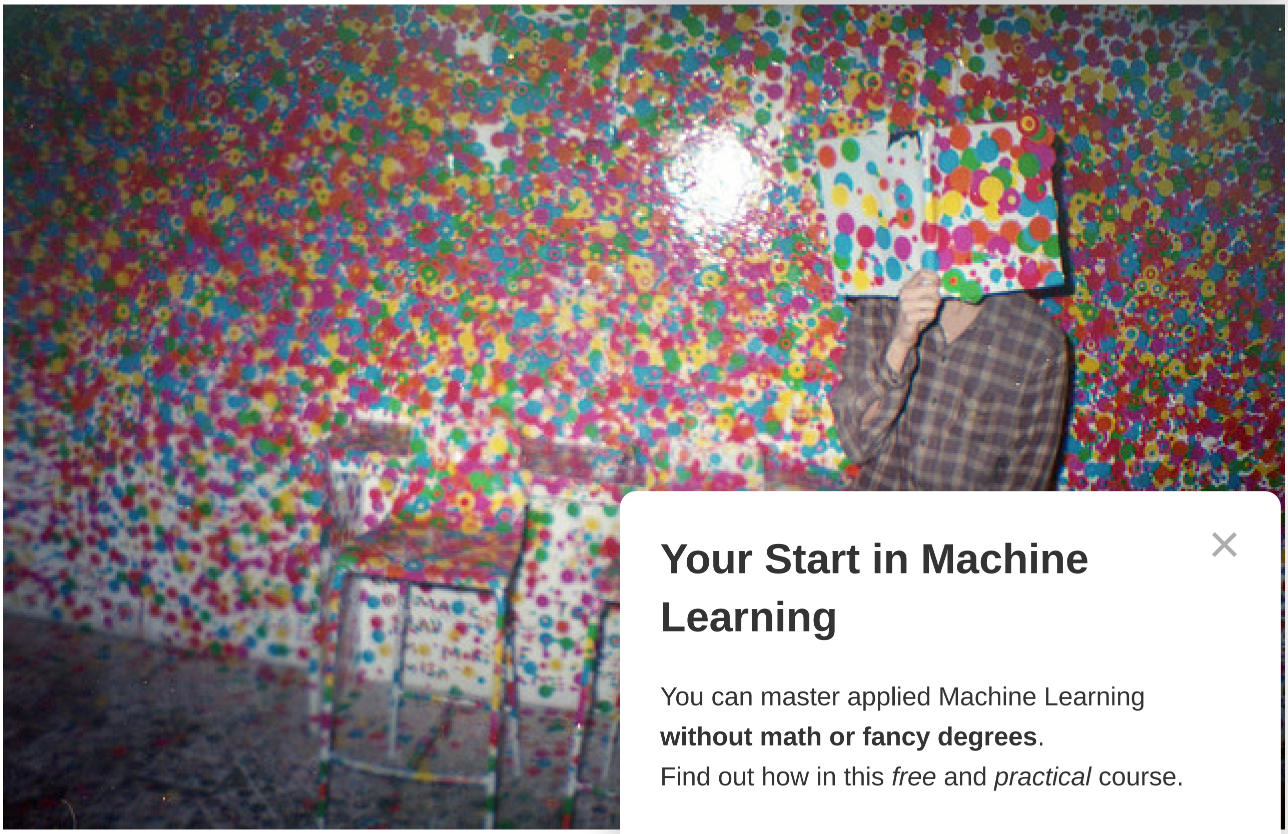
- How to disentangle the many names used to refer to support vector machines.
- The representation used by SVM when the model is actually stored on disk.
- How a learned SVM model representation can be used to make predictions for new data.
- How to learn an SVM model from training data.
- How to best prepare your data for the SVM algorithm.
- Where you might look to get more information on SVM.

SVM is an exciting algorithm and the concepts are relatively simple. This post was written for developers with little or no background in statistics and linear algebra.

As such we will stay high-level in this description and focus on the specific implementation concerns. The question around why specific equations are used or how they were derived are not covered and you may want to dive deeper in the further reading section.

Discover how machine learning algorithms work including kNN, decision trees, naive bayes, SVM, ensembles and much more in my new book, with 22 tutorials and examples in excel.

Let's get started.



Support Vector Machi
Photo by Francisco Bar

## Maximal-Margin Classifier

The Maximal-Margin Classifier is a hypothetical cla ice.

The numeric input variables (x) in your data (the co e, if you had two input variables, this would form a two-dimensional space.

A hyperplane is a line that splits the input variable space. In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-dimensions you can visualize this as a line and let's assume that all of our input points can be completely separated by this line. For example:

$$B0 + (B1 * X1) + (B2 * X2) = 0$$

Where the coefficients (B1 and B2) that determine the slope of the line and the intercept (B0) are found by the learning algorithm, and X1 and X2 are the two input variables.

You can make classifications using this line. By plugging in input values into the line equation, you can calculate whether a new point is above or below the line.

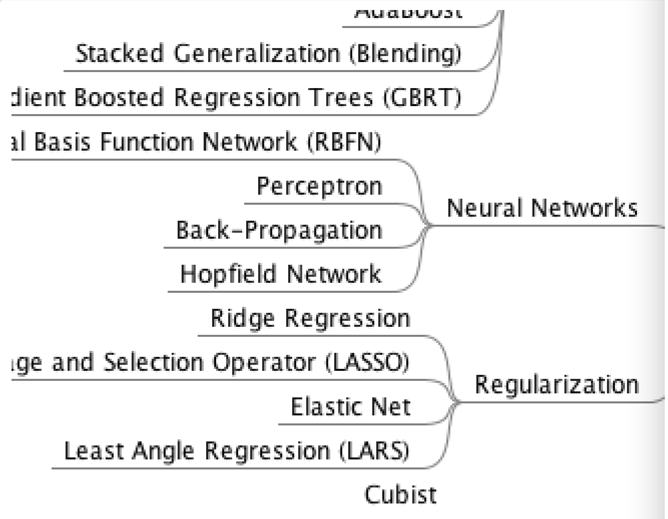
- Above the line, the equation returns a value greater than 0 and the point belongs to the first class (class 0).
- Below the line, the equation returns a value less than 0 and the point belongs to the second class (class 1).
- A value close to the line returns a value close to zero and the point may be difficult to classify.
- If the magnitude of the value is large, the mod

The distance between the line and the closest data points is referred to as the margin. The best or optimal line that can separate the two classes is the line that as the largest margin. This is called the Maximal-Margin hyperplane.

The margin is calculated as the perpendicular distance from the line to only the closest points. Only these points are relevant in defining the line and in the construction of the classifier. These points are called the support vectors. They support or define the hyperplane.

The hyperplane is learned from training data using an optimization procedure that maximizes the margin.

## Get your FREE Algorithms Mind Map


AdaBoost
Stacked Generalization (Blending)
dient Boosted Regression Trees (GBRT)
al Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Neural Networks
Ridge Regression
age and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Regularization
Cubist

Sample of the handy machine learning algorithms mind

ine

se.

## Soft Margin Classifier

In practice, real data is messy and cannot be separated perfectly with a hyperplane.

The constraint of maximizing the margin of the line that separates the classes must be relaxed. This is often called the soft margin classifier. This change allows some points in the training data to violate the separating line.

An additional set of coefficients are introduced that give the margin wiggle room in each dimension. These coefficients are sometimes called slack variables. This increases the complexity of the model as there are more parameters for the model to fit to the data to provide this complexity.

A tuning parameter is introduced called simply C that defines the magnitude of the wiggle allowed across all dimensions. The C parameters defines the amount of violation of the margin allowed. A C=0 is no violation and we are back to the inflexible Maximal-Margin Classifier described above. The larger the value of C the more violations of the hyperplane are permitted.

During the learning of the hyperplane from data, all training instances that lie within the distance of the margin will affect the placement of the hyperplane and are referred to as support vectors. And as C affects the number of instances that are allowed to fall within the margin, C influences the number of support vectors used by the model.

- The smaller the value of C, the more sensitive the algorithm is to the training data (higher variance and lower bias).
- The larger the value of C, the less sensitive the algorithm is to the training data (lower variance and higher bias).

## Support Vector Machines (Kernels)

The SVM algorithm is implemented in practice using a kernel.

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra, which is out of the scope of this introduction to SVM.

A powerful insight is that the linear SVM can be re
observations, rather than the observations themse e
sum of the multiplication of each pair of input value

For example, the inner product of the vectors [2, 3

The equation for making a prediction for a new inp
each support vector (xi) is calculated as follows:

$$f(x) = B0 +$$

This is an equation that involves calculating the inn ort
vectors in training data. The coefficients B0 and ai ng
data by the learning algorithm.

### Linear Kernel SVM

The dot-product is called the kernel and can be re-written as:

$$K(x, xi) = sum(x * xi)$$

The kernel defines the similarity or a distance measure between new data and the support vectors. The dot product is the similarity measure used for linear SVM or a linear kernel because the distance is a linear combination of the inputs.

Other kernels can be used that transform the input space into higher dimensions such as a Polynomial Kernel and a Radial Kernel. This is called the Kernel Trick.

It is desirable to use more complex kernels as it allows lines to separate the classes that are curved or even more complex. This in turn can lead to more accurate classifiers.

### Polynomial Kernel SVM

Instead of the dot-product, we can use a polynomial kernel, for example:

$$K(x,xi) = 1 + sum(x * xi)^d$$

Where the degree of the polynomial must be specified by hand to the learning algorithm. When d=1 this is the same as the linear kernel. The polynomial kernel allows for curved lines in the input space.

### Radial Kernel SVM

Finally, we can also have a more complex radial kernel. For example:

K(x,xi) = exp(-gamma * sum((x – xi^2))

Where gamma is a parameter that must be specified to the learning algorithm. A good default value for gamma is 0.1, where gamma is often 0 < gamma < 1. The radial kernel is very local and can create complex regions within the feature space, like closed polygons in two-dimensional space.

## How to Learn a SVM Model

The SVM model needs to be solved using an optin

You can use a numerical optimization procedure to s is
inefficient and is not the approach used in widely u
implementing the algorithm as an exercise, you co

There are specialized optimization procedures that
Quadratic Programming problem. The most popula l
Optimization (SMO) method that is very efficient. It can
be solved analytically (by calculating) rather than r

## Data Preparation for SVM

This section lists some suggestions for how to best prepare your training data when learning an SVM model.

- **Numerical Inputs**: SVM assumes that your inputs are numeric. If you have categorical inputs you may need to covert them to binary dummy variables (one variable for each category).
- **Binary Classification**: Basic SVM as described in this post is intended for binary (two-class) classification problems. Although, extensions have been developed for regression and multi-class classification.

## Further Reading

Support Vector Machines are a huge area of study. There are numerous books and papers on the topic. This section lists some of the seminal and most useful results if you are looking to dive deeper into the background and theory of the technique.

Vladimir Vapnik, one of the inventors of the technique has two books that are considered seminal on the topic. They are very mathematical and also rigorous.

- The Nature of Statistical Learning Theory, Vapnik, 1995
- Statistical Learning Theory, Vapnik, 1998

Any good book on machine learning will cover SVM, below are some of my favorites.

- An Introduction to Statistical Learning: with Applications in R, Chapter 8
- Applied Predictive Modeling, Chapter 13
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction Chapter 12

There are countless tutorials and journal articles on SVM. Below is a link to a seminal paper on SVM by Cortes and Vapnik and another to an excellent introductory tutorial.

- Support-Vector Networks [PDF] by Cortes and Vapnik 1995
- A Tutorial on Support Vector Machines for Pattern Recognition [PDF] 1998

Wikipedia provides some good (although dense) in

- Support Vector Machine on Wikipedia
- Wikibook on Support Vector Machines

Finally, there are a lot of posts on Q&A sites asking picks that you might find useful.

- What does support vector machine (SVM) me
- Please explain Support Vector Machines (SV

## Summary

In this post you discovered the Support Vector Mac d about:

- The Maximal-Margin Classifier that provides a
- The Soft Margin Classifier which is a modification of the Maximal-Margin Classifier to relax the margin to handle noisy class boundaries in real data.
- Support Vector Machines and how the learning algorithm can be reformulated as a dot-product kernel and how other kernels like Polynomial and Radial can be used.
- How you can use numerical optimization to learn the hyperplane and that efficient implementations use an alternate optimization scheme called Sequential Minimal Optimization.

Do you have any questions about SVM or this post?
Ask in the comments and I will do my best to answer.