# Support-vector machine

From Wikipedia, the free encyclopedia

In machine learning, **support-vector machines (SVMs,** also **support-vector networks**[1]) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Cortes and Vapnik, 1995,[1] Vapnik et al., 1997[citation needed]), SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik (1982, 1995) and Chervonenkis (1974). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

SVM machine learning

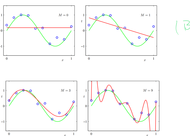
# Bias–variance tradeoff

From Wikipedia, the free encyclopedia

In statistics and machine learning, the **bias–variance tradeoff** is the property of a model that the variance of the parameter estimated across samples can be reduced by increasing bias in the estimated parameters. The **bias–variance dilemma** or **bias–variance problem** is the conflict in trying to simultaneously minimize these two sources of error that prevent supervised learning algorithms from generalizing beyond their training set:[1][2]

- The bias error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- The variance is an error from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

The **bias–variance decomposition** is a way of analyzing a learning algorithm's expected generalization error with respect to a particular problem as a sum of three terms, the bias, variance, and a quantity called the *irreducible error*, resulting from noise in the problem itself.




(Bishop)

**Figure 1.4** Plots of polynomials having various orders M, shown as red curves, fitted to the data set shown in Figure 1.2.

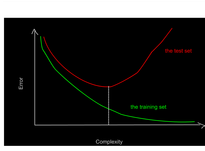
$$y_i = f(x_i) + \varepsilon_i$$

$$\hat{y}_i = \hat{f}(x_i)$$

$$E[(\hat{y} - y)^2] = \ldots\ldots$$

$$Err = \sigma_\varepsilon - m\sigma^2 + 2\sigma^2 \sum_{i=1}^{m} D_i$$

$$D_i = \frac{\partial \hat{f}}{\partial y_i}$$




Cross-Validation

(NRK)

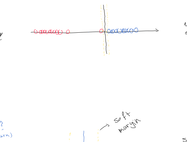
**Cross-validation**,[1][2][3] sometimes called **rotation estimation**[4][5][6][7] or **out-of-sample testing**, is any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set. Cross-validation is a resampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of *known data* on which training is run (*training dataset*), and a dataset of *unknown data* (or *first seen* data) against which the model is tested (called the validation dataset or *testing set*).[8][9] The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias[10] and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

(Bishop)

**Figure 1.18** The technique of 5-fold cross-validation, illustrated here for the case of $S = 4$, involves taking the available data and partitioning it into $S$ groups (in the simplest case these are of equal size). Then $S - 1$ of the groups are used to train a set of models that are then evaluated on the remaining group. This procedure is then repeated for all $S$ possible choices for the held-out group, indicated here by the red blocks, and the performance scores from the $S$ runs are then averaged.

an example of classification $x$: weights

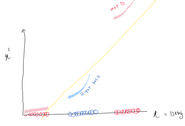
B → diabetes
R → not Diabetes




sailing down off street?

very sensitive to outliers

hyperparameter? C-2 (Uniform)



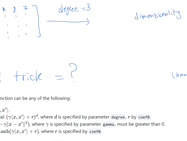

Some mistake Rather allowed

How to find soft margins? { cross-validation / Bias-variance minimization

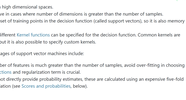
now a little different situation.

It is hardly separable.

Hard to find a proper separator




2D example: we have data points to classify by hand models.




A hyperplane can not separate the data

Polynomial feature extraction?







Kernel trick = ?

(kernel (x,x)?

The *kernel function* can be any of the following:

- linear: $\langle x, x' \rangle$.
- polynomial: $(\gamma \langle x, x' \rangle + r)^d$, where $d$ is specified by parameter `degree`, $r$ by `coef0`.
- rbf: $\exp(-\|x - x'\|^2)$, where $\gamma$ is specified by parameter `gamma`, must be greater than 0.
- sigmoid $\tanh(\gamma \langle x, x' \rangle + r)$, where $r$ is specified by `coef0`.

The advantages of support vector machines are:

(Sklearn)

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).