

# 支持向量机

维基百科，自由的百科全书

在机器学习中，**支持向量机**（英語：**support vector machine**，常简称为**SVM**，又名**支持向量网络**<sup>[1]</sup>）是在分类与迴歸分析中分析数据的**監督式學習**模型与相关的学习算法。给定一组训练实例，每个训练实例被标记为属于两个类别中的一个或另一个，SVM训练算法建立一个将新的实例分配给两个类别之一的模型，使其成为非概率二元线性分类器。SVM模型是将实例表示为空间中的点，这样映射就使得单独类别的实例被尽可能宽的明显的间隔分开。然后，将新的实例映射到同一空间，并基于它们落在间隔的哪一侧来预测所属类别。

除了进行线性分类之外，SVM还可以使用所谓的**核技巧**有效地进行非线性分类，将其输入隐式映射到高维特征空间中。

当数据未被标记时，不能进行监督式学习，需要用非**監督式學習**，它会尝试找出数据到簇的自然聚类，并将新数据映射到这些已形成的簇。将支持向量机改进的聚类算法被称为**支持向量聚类**<sup>[2]</sup>，当数据未被标记或者仅一些数据被标记时，支持向量聚类经常在工业应用中用作分类步骤的预处理。

# 目录

---

[动机](#)

[定义](#)

[应用](#)

[历史](#)

[线性SVM](#)

[硬间隔](#)

[软间隔](#)

[非线性分类](#)

[计算SVM分类器](#)

[原型](#)

[对偶型](#)

[核技巧](#)

[现代方法](#)

[次梯度下降](#)

[坐标下降](#)

[性质](#)

[参数选择](#)

[问题](#)

[延伸](#)

[实现](#)

[参见](#)

[参考文献](#)

[引用](#)

[来源](#)

[外部链接](#)

## 动机

---

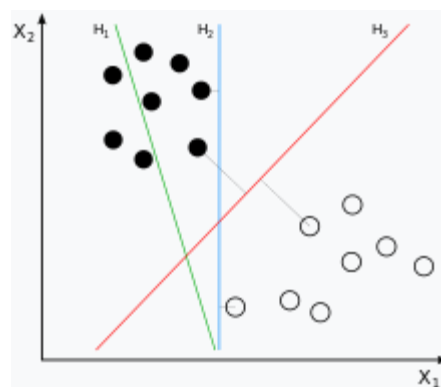
将数据进行分类是机器学习中的一项常见任务。假设某些给定的数据点各自属于两个类之一，而目标是确定新数据点将在哪个类中。对于支持向量机来说，数据点被视为  $p$  维向量，而我们想知道是否可以用  $(p - 1)$  维超平面来分开这些点。这就是所谓的线性分类器。可能有许多超平面可以把数据分类。最佳超平面的一个合理选择是以最大间隔把两个类分开的超平面。因此，我们要选择能够

让到每边最近的数据点的距离最大化的超平面。如果存在这样的超平面，则称为最大间隔超平面，而其定义的线性分类器被称为最大间隔分类器，或者叫做最佳稳定性感知器。

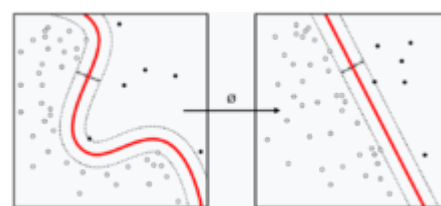
## 定义

更正式地来说，支持向量机在高维或无限维空间中构造超平面或超平面集合，其可以用于分类、回归或其他任务。直觀來說，分類邊界距離最近的訓練資料點越遠越好，因為這樣可以縮小分類器的泛化誤差。

尽管原始问题可能是在有限维空间中陈述的，但用于区分的集合在该空间中往往线性不可分。为此，有人提出将原有限维空间映射到维数高得多的空间中，在该空间中进行分离可能会更容易。为了保持计算负荷合理，人们选择适合该问题的核函数  $k(x, y)$  来定义SVM方案使用的映射，以确保用原始空间中的变量可以很容易计算点积。<sup>[3]</sup> 高维空间中的超平面定义为与该空间中的某向量的点积是常数的点的集合。定义超平面的向量可以选择在数据基中出现的特征向量  $x_i$  的图像的参数  $\alpha_i$  的线性组合。通过选择超平面，被映射到超平面上的特征空间中的点集  $x$  由以下关系定义： $\sum_i \alpha_i k(x_i, x) = \text{constant}$ . 注意，如果随着  $y$  逐渐远离  $x$ ， $k(x, y)$  变小，则求和中的每一项都是在衡量测试点  $x$  与对应的数据基点  $x_i$  的接近程度。这样，上述内核的总和可以用于衡量每个测试点相对于待分离的集合中的数据点的相对接近度。



H<sub>1</sub> 不能把类别分开。H<sub>2</sub> 可以，但只有很小的间隔。H<sub>3</sub> 以最大间隔将它们分开。



核机器

## 应用

- 用于文本和超文本的分类，在归纳和直推方法中都可以显著减少所需要的有类标的样本数。
- 用于图像分类。实验结果显示：在经过三到四轮相关反馈之后，比起传统的查询优化方案，支持向量机能够取得明显更高的搜索准确度。这同样也适用于图像分割系统，比如使用Vapnik所建议的使用特权方法的修改版本SVM的那些图像分割系统。<sup>[4][5]</sup>
- 用于手写字体识别。
- 用于医学中分类蛋白质，超过90%的化合物能够被正确分类。基于支持向量机权重的置换测试已被建议作为一种机制，用于解释的支持向量机模型。<sup>[6][7]</sup> 支持向量机权重也被用来解释过去的SVM模型。<sup>[8]</sup> 为识别模型用于进行预测的特征而对支持向量机模型做出事后解释是在生物科学中具有特殊意义的相对较新的研究领域。

## 历史

原始SVM算法是由弗拉基米尔·万普尼克和亞歷克塞·澤范蘭傑斯于1963年发明的。1992年，Bernhard E. Boser、Isabelle M. Guyon和弗拉基米尔·万普尼克提出了一种通过将核技巧应用于最大间隔超平面来创建非线性分类器的方法。<sup>[9]</sup> 当前标准的前身（软间隔）由Corinna Cortes和Vapnik于1993年提出，并于1995年发表。<sup>[1]</sup>

## 线性SVM

我们考虑以下形式的  $n$  点测试集：

$$(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$$

其中  $y_i$  是 1 或者 -1，表明点  $\vec{x}_i$  所属的类。 $\vec{x}_i$  中每个都是一个  $p$  维实向量。我们要求将  $y_i = 1$  的点集  $\vec{x}_i$  与  $y_i = -1$  的点集分开的“最大间隔超平面”，使得超平面与最近的点  $\vec{x}_i$  之间的距离最大化。

任何超平面都可以写作满足下面方程的点集  $\vec{x}$

$$\vec{w} \cdot \vec{x} - b = 0,$$

其中  $\vec{w}$ （不必是归一化的）是该法向量。参数  $\frac{b}{\|\vec{w}\|}$  决定从原点沿法向量  $\vec{w}$  到超平面的偏移量。

## 硬间隔

如果这些训练数据是线性可分的，可以选择分离两类数据的两个平行超平面，使得它们之间的距离尽可能大。在这两个超平面范围内的区域称为“间隔”，最大间隔超平面是位于它们正中间的超平面。这些超平面可以由方程：

$$\vec{w} \cdot \vec{x} - b = 1$$

或是

$$\vec{w} \cdot \vec{x} - b = -1.$$

来表示。通过几何不难得到这两个超平面之间的距离是  $\frac{2}{\|\vec{w}\|}$ ，因此要使两平面间的距离最大，我们需要最小化  $\|\vec{w}\|$ 。同时为了使得样本数据点都在超平面的间隔区以外，我们需要保证对于所有的  $i$  满足其中的一个条件：

$$\vec{w} \cdot \vec{x}_i - b \geq 1, \text{ 若 } y_i = 1$$

或是

$$\vec{w} \cdot \vec{x}_i - b \leq -1, \text{ 若 } y_i = -1.$$

这些约束表明每个数据点都必须位于间隔的正确一侧。

这两个式子可以写作：

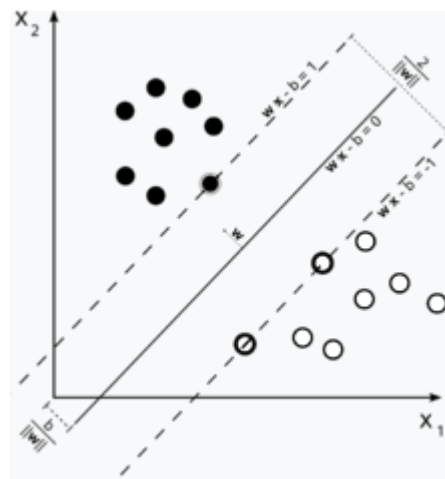
$$y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq n. \quad (1)$$

可以用这个式子一起来得到优化问题：

$$\left| \begin{array}{l} \text{“在 } y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1 \text{ 条件下，最小化 } \|\vec{w}\|, \text{ 对于 } i = 1, \dots, n \text{”} \end{array} \right.$$

这个问题的解  $\vec{w}$  与  $b$  决定了我们的分类器  $\vec{x} \mapsto \text{sgn}(\vec{w} \cdot \vec{x} - b)$ 。

此几何描述的一个显而易见却重要的结果是，最大间隔超平面完全是由最靠近它的那些  $\vec{x}_i$  确定的。这些  $\vec{x}_i$  叫做**支持向量**。



设样本属于两个类，用该样本训练 SVM 得到的最大间隔超平面。在边缘上的样本点称为支持向量。

## 软间隔

为了将SVM扩展到数据线性不可分的情况，我们引入铰链损失函数，

$$\max \left( 0, 1 - y_i (\vec{w} \cdot \vec{x}_i - b) \right).$$

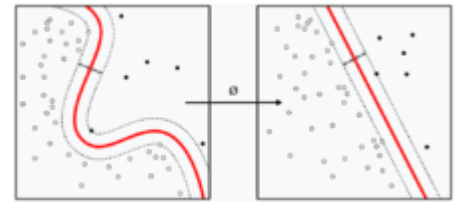
当约束条件 (1) 满足时（也就是如果  $\vec{x}_i$  位于边距的正确一侧）此函数为零。对于间隔的错误一侧的数据，该函数的值与距间隔的距离成正比。然后我们希望最小化

$$\left[ \frac{1}{n} \sum_{i=1}^n \max \left( 0, 1 - y_i (\vec{w} \cdot \vec{x}_i - b) \right) \right] + \lambda \|\vec{w}\|^2,$$

其中参数  $\lambda$  用来权衡增加间隔大小与确保  $\vec{x}_i$  位于间隔的正确一侧之间的关系。因此，对于足够小的  $\lambda$  值，如果输入数据是可以线性分类的，则软间隔SVM与硬间隔SVM将表现相同，但即使不可线性分类，仍能学习出可行的分类规则。

## 非线性分类

万普尼克在1963年提出的原始最大间隔超平面算法构造了一个线性分类器。而1992年，Bernhard E. Boser、Isabelle M. Guyon和弗拉基米尔·万普尼克提出了一种通过将核技巧（最初由Aizerman et al.<sup>[10]</sup>提出）应用于最大边界超平面来创建非线性分类器的方法。<sup>[11]</sup> 所得到的算法形式上类似，除了把点积换成了非线性核函数。这就允许算法在变换后的特征空间中拟合最大间隔超平面。该变换可以是非线性的，而变换空间是高维的；虽然分类器是变换后的特征空间中的超平面，但它在原始输入空间中可以是非线性的。



核机器

值得注意的是，更高维的特征空间增加了支持向量机的泛化误差，但给定足够多的样本，算法仍能表现良好。<sup>[12]</sup>

常见的核函数包括：

- 齐次多项式： $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j)^d$
- 非齐次多项式： $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d$
- 高斯径向基函数： $k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2)$ ，其中  $\gamma > 0$ 。有时参数化表示  $\gamma = 1/2\sigma^2$
- 双曲正切： $k(\vec{x}_i, \vec{x}_j) = \tanh(\kappa \vec{x}_i \cdot \vec{x}_j + c)$ ，其中一些（而非所有） $\kappa > 0$  且  $c < 0$

由等式  $k(\vec{x}_i, \vec{x}_j) = \varphi(\vec{x}_i) \cdot \varphi(\vec{x}_j)$ ，核函数与变换  $\varphi(\vec{x}_i)$  有关。变换空间中也有  $\vec{w}$  值， $\vec{w} = \sum_i \alpha_i y_i \varphi(\vec{x}_i)$ 。与  $\vec{w}$  的点积也要用核技巧来计算，即  $\vec{w} \cdot \varphi(\vec{x}) = \sum_i \alpha_i y_i k(\vec{x}_i, \vec{x})$ 。

## 计算SVM分类器

计算（软间隔）SVM分类器等同于使下面表达式最小化

$$\left| \left[ \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i(w \cdot x_i + b)) \right] + \lambda \|w\|^2. \quad (2) \right.$$

如上所述，由于我们关注的是软间隔分类器， $\lambda$  选择足够小的值就能得到线性可分类输入数据的硬间隔分类器。下面会详细介绍将(2)简化为二次规划问题的经典方法。之后会讨论一些最近才出现的方法，如次梯度下降法和坐标下降法。

## 原型

最小化(2)可以用下面的方式改写为目标函数可微的约束优化问题。

对所有  $i \in \{1, \dots, n\}$  我们引入变量  $\zeta_i = \max(0, 1 - y_i(w \cdot x_i + b))$ 。注意到  $\zeta_i$  是满足  $y_i(w \cdot x_i + b) \geq 1 - \zeta_i$  的最小非负数。

因此，我们可以将优化问题叙述如下

$$\left| \begin{array}{l} \text{minimize } \frac{1}{n} \sum_{i=1}^n \zeta_i + \lambda \|w\|^2 \\ \text{subject to } y_i(x_i \cdot w + b) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0, \text{ for all } i. \end{array} \right.$$

这就叫做原型问题。

## 对偶型

通过求解上述问题的拉格朗日对偶，得到简化的问题

$$\left| \begin{array}{l} \text{maximize } f(c_1 \dots c_n) = \sum_{i=1}^n c_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i c_i (x_i \cdot x_j) y_j c_j, \\ \text{subject to } \sum_{i=1}^n c_i y_i = 0, \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda} \text{ for all } i. \end{array} \right.$$

这就叫做对偶问题。由于对偶最小化问题是受线性约束的  $c_i$  的二次函数，所以它可以通过二次规划算法高效地解出。这里，变量  $c_i$  定义为满足

$$\left| \vec{w} = \sum_{i=1}^n c_i y_i \vec{x}_i. \right.$$

此外，当  $\vec{x}_i$  恰好在间隔的正确一侧时  $c_i = 0$ ，且当  $\vec{x}_i$  位于间隔的边界时  $0 < c_i < (2n\lambda)^{-1}$ 。因此， $\vec{w}$  可以写为支持向量的线性组合。可以通过在间隔的边界上找到一个  $\vec{x}_i$  并求解

$$\left| y_i(\vec{w} \cdot \vec{x}_i + b) = 1 \iff b = y_i - \vec{w} \cdot \vec{x}_i. \right.$$

得到偏移量  $b$ 。（注意由于  $y_i = \pm 1$  因而  $y_i^{-1} = y_i$ 。）

## 核技巧

假设我们要学习与变换后数据点  $\varphi(\vec{x}_i)$  的线性分类规则对应的非线性分类规则。此外，我们有一个满足  $k(\vec{x}_i, \vec{x}_j) = \varphi(\vec{x}_i) \cdot \varphi(\vec{x}_j)$  的核函数  $k$ 。

我们知道变换空间中的分类向量  $\vec{w}$  满足

$$\vec{w} = \sum_{i=1}^n c_i y_i \varphi(\vec{x}_i),$$

其中  $c_i$  可以通过求解优化问题

$$\begin{aligned} \text{maximize } f(c_1 \dots c_n) &= \sum_{i=1}^n c_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i c_i (\varphi(\vec{x}_i) \cdot \varphi(\vec{x}_j)) y_j c_j \\ &= \sum_{i=1}^n c_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i c_i k(\vec{x}_i, \vec{x}_j) y_j c_j \\ \text{subject to } \sum_{i=1}^n c_i y_i &= 0, \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda} \text{ for all } i. \end{aligned}$$

得到。与前面一样，可以使用二次规划来求解系数  $c_i$ 。同样，我们可以找到让  $0 < c_i < (2n\lambda)^{-1}$  的索引  $i$ ，使得  $\varphi(\vec{x}_i)$  位于变换空间中间隔的边界上，然后求解

$$\begin{aligned} b = \vec{w} \cdot \varphi(\vec{x}_i) - y_i &= \left[ \sum_{k=1}^n c_k y_k \varphi(\vec{x}_k) \cdot \varphi(\vec{x}_i) \right] - y_i \\ &= \left[ \sum_{k=1}^n c_k y_k k(\vec{x}_k, \vec{x}_i) \right] - y_i. \end{aligned}$$

最后，可以通过计算下式来分类新点

$$\vec{z} \mapsto \text{sgn}(\vec{w} \cdot \varphi(\vec{z}) + b) = \text{sgn}\left(\left[\sum_{i=1}^n c_i y_i k(\vec{x}_i, \vec{z})\right] + b\right).$$

## 现代方法

用于找到SVM分类器的最近的算法包括次梯度下降和坐标下降。当处理大的稀疏数据集时，这两种技术已经被证明有着显著的优点——当存在许多训练实例时次梯度法是特别有效的，并且当特征空间的维度高时，坐标下降特别有效。

### 次梯度下降

SVM的次梯度下降算法直接用表达式

$$f(\vec{w}, b) = \left[ \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i (w \cdot x_i + b)) \right] + \lambda \|w\|^2.$$

注意  $f$  是  $\vec{w}$  与  $b$  的凸函数。因此，可以采用传统的梯度下降（或SGD）方法，其中不是在函数梯度的方向上前进，而是在从函数的次梯度中选出的向量的方向上前进。该方法的优点在于，对于某些实现，迭代次数不随着数据点的数量  $n$  而增加或减少。<sup>[13]</sup>

## 坐标下降

SVM的坐标下降算法基于对偶问题

$$\begin{aligned} \text{maximize } f(c_1 \dots c_n) &= \sum_{i=1}^n c_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i c_i (x_i \cdot x_j) y_j c_j, \\ \text{subject to } \sum_{i=1}^n c_i y_i &= 0, \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda} \text{ for all } i. \end{aligned}$$

对所有  $i \in \{1, \dots, n\}$  进行迭代，使系数  $c_i$  的方向与  $\partial f / \partial c_i$  一致。然后，将所得的系数向量  $(c'_1, \dots, c'_n)$  投影到满足给定约束的最接近的系数向量。（通常使用欧氏距离。）然后重复该过程，直到获得接近最佳的系数向量。所得的算法在实践中运行非常快，尽管已经证明的性能保证很少。<sup>[14]</sup>

## 性质

SVM属于广义线性分类器的一族，并且可以解释为感知器的延伸。它们也可以被认为是吉洪诺夫正则化的特例。它们有一个特别的性质，就是可以同时最小化经验误差和最大化几何边缘区；因此它们也被称为**最大间隔分类器**。

Meyer、Leisch和Hornik对SVM与其他分类器进行了比较。<sup>[15]</sup>

## 参数选择

SVM的有效性取决于核函数、核参数和软间隔参数  $C$  的选择。通常会选只有一个参数  $\gamma$  的高斯核。 $C$  和  $\gamma$  的最佳组合通常通过在  $C$  和  $\gamma$  为指数增长序列下网格搜索来选取，例如  $C \in \{2^{-5}, 2^{-3}, \dots, 2^{13}, 2^{15}\}$ ;  $\gamma \in \{2^{-15}, 2^{-13}, \dots, 2^1, 2^3\}$ 。通常情况下，使用交叉验证来检查参数选择的每一个组合，并选择具有最佳交叉验证精度的参数。或者，最近在贝叶斯优化中的工作可以用于选择  $C$  和  $\gamma$ ，通常需要评估比网格搜索少得多的参数组合。或者，贝叶斯优化的最近进展可以用于选择  $C$  和  $\gamma$ ，通常需要计算的参数组合比网格搜索少得多。然后，使用所选择的参数在整个训练集上训练用于测试和分类新数据的最终模型。<sup>[16]</sup>

## 问题

SVM的潜在缺点包括以下方面：

- 需要对输入数据进行完全标记
- 未校准类成员概率
- SVM仅直接适用于两类任务。因此，必须应用将多类任务减少到几个二元问题的算法；请参阅多类SVM一节。



- 解出的模型的参数很难理解。

## 延伸

---

- 支持向量聚類：支持向量聚類是一種建立在核函數上的類似方法，同適用於非監督學習和數據挖掘。它被認為是數據科學中的一種基本方法。
- 轉導支持向量機
- 多元分類支持向量機：SVM算法最初是為二值分類問題設計的，實現多分類的主要方法是將一個多分類問題轉化為多個二分類問題。常見方法包括“一對多法”和“一對一法”，一對多法是將某個類別的樣本歸為一類，其他剩餘的樣本歸為另一類，這樣k個類別的樣本就構造出了k個二分類SVM；一對一法則是在任意兩類樣本之間設計一個SVM。
- 支持向量回歸
- 結構化支持向量機：支持向量機可以被推廣為結構化的支持向量機，推廣後標籤空間是結構化的並且可能具有無限的大小。

## 实现

---

最大间隔超平面的参数是通过求解优化得到的。有几种专门的算法可用于快速解决由SVM产生的QP问题，它们主要依靠启发式算法将问题分解成更小、更易于处理的子问题。

另一种方法是使用内点法，其使用类似牛顿法的迭代找到卡羅需－庫恩－塔克條件下原型和对偶型的解。<sup>[17]</sup> 这种方法不是去解决一系列分解问题，而是直接完全解决该问题。为了避免求解核矩阵很大的线性系统，在核技巧中经常使用矩阵的低秩近似。

另一个常见的方法是普莱特的序列最小优化算法（SMO），它把问题分成了若干个可以解析求解的二维子问题，这样就可以避免使用数值优化算法和矩阵存储。<sup>[18]</sup>

线性支持向量机的特殊情况可以通过用于优化其类似问题邏輯迴歸的同类算法更高效求解；这类算法包括次梯度下降法（如PEGASOS<sup>[19]</sup>）和坐标下降法（如LIBLINEAR<sup>[20]</sup>）。LIBLINEAR有一些引人注目的训练时间上的特性。每次收敛迭代花费在读取训练数据上的时间是线性的，而且这些迭代还具有Q-线性收敛特性，使得算法非常快。

一般的核SVM也可以用次梯度下降法（P-packSVM<sup>[21]</sup>）更快求解，在允许并行化时求解速度尤其快。

许多机器学习工具包都可以使用核SVM，有LIBSVM、MATLAB、SAS (<http://support.sas.com/documentation/cdl/en/whatsnew/64209/HTML/default/viewer.htm#emdocwhatsnew71.htm>)、SVMlight、kernlab (<https://cran.r-project.org/package=kernlab>)、scikit-learn、Shogun、Weka、Shark (<http://image.diku.dk/shark/>)、JKernelMachines (<https://mloss.org/software/view/409/>)、OpenCV等。

## 参见

---

- 核机器
- 费希尔核
- 多项式核函数

- 预测分析
- 相关向量机，函数形式与SVM相同的概率稀疏核模型
- 序列最小优化算法
- 空间映射

## 参考文献

---

### 引用

1. Cortes, C.; Vapnik, V. Support-vector networks. *Machine Learning*. 1995, **20** (3): 273–297. doi:10.1007/BF00994018.
2. Ben-Hur, Asa, Horn, David, Siegelmann, Hava, and Vapnik, Vladimir; "Support vector clustering" (2001) *Journal of Machine Learning Research*, 2: 125–137.
3. \*Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, B. P. Section 16.5. Support Vector Machines. *Numerical Recipes: The Art of Scientific Computing 3rd*. New York: Cambridge University Press. 2007. ISBN 978-0-521-88068-8.
4. Vapnik, V.: Invited Speaker. IPMU Information Processing and Management 2014)
5. Barghout, Lauren. "Spatial-Taxon Information Granules as Used in Iterative Fuzzy-Decision-Making for Image Segmentation." *Granular Computing and Decision-Making*. Springer International Publishing, 2015. 285-318.
6. Bilwaj Gaonkar, Christos Davatzikos Analytic estimation of statistical significance maps for support vector machine based multi-variate image analysis and classification
7. R. Cuingnet, C. Rosso, M. Chupin, S. Lehericy, D. Dormont, H. Benali, Y. Samson and O. Colliot, Spatial regularization of SVM for the detection of diffusion alterations associated with stroke outcome, *Medical Image Analysis*, 2011, 15 (5): 729–737
8. Statnikov, A., Hardin, D., & Aliferis, C. (2006). Using SVM weight-based methods to identify causally relevant and non-causally relevant variables. *sign*, 1, 4.
9. Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory – COLT '92*. 1992: 144. ISBN 089791497X. doi:10.1145/130385.130401.
10. Aizerman, Mark A.; Braverman, Emmanuel M. & Rozonoer, Lev I. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*. 1964, **25**: 821–837.
11. Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory – COLT '92*. 1992: 144. ISBN 089791497X. doi:10.1145/130385.130401.
12. Jin, Chi; Wang, Liwei. Dimensionality dependent PAC-Bayes margin bound. *Advances in Neural Information Processing Systems*. 2012.
13. Shalev-Shwartz, Shai; Singer, Yoram; Srebro, Nathan; Cotter, Andrew. Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming*. 2010-10-16, **127** (1): 3–30. ISSN 0025-5610. doi:10.1007/s10107-010-0420-4.
14. Hsieh, Cho-Jui; Chang, Kai-Wei; Lin, Chih-Jen; Keerthi, S. Sathiya; Sundararajan, S. A Dual Coordinate Descent Method for Large-scale Linear SVM. *Proceedings of the 25th International Conference on Machine Learning. ICML '08* (New York, NY, USA: ACM). 2008-01-01: 408–415. ISBN 978-1-60558-205-4. doi:10.1145/1390156.1390208.
15. Meyer, D.; Leisch, F.; Hornik, K. The support vector machine under test. *Neurocomputing*. 2003, **55**: 169. doi:10.1016/S0925-2312(03)00431-4.

16. Hsu, Chih-Wei; Chang, Chih-Chung & Lin, Chih-Jen. A Practical Guide to Support Vector Classification (PDF) (Technical report). Department of Computer Science and Information Engineering, National Taiwan University. 2003.
17. Ferris, M. C.; Munson, T. S. Interior-Point Methods for Massive Support Vector Machines. *SIAM Journal on Optimization*. 2002, **13** (3): 783. doi:10.1137/S1052623400374379.
18. John C. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines (PDF). NIPS. 1998.
19. Shai Shalev-Shwartz; Yoram Singer; Nathan Srebro. Pegasos: Primal Estimated sub-GrAdient SOLver for SVM (PDF). ICML. 2007.
20. R.-E. Fan; K.-W. Chang; C.-J. Hsieh; X.-R. Wang; C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*. 2008, **9**: 1871–1874.
21. Zeyuan Allen Zhu; 等. P-packSVM: Parallel Primal grAdient desCent Kernel SVM (PDF). ICDM. 2009.

## 来源

- Theodoridis, Sergios; and Koutroumbas, Konstantinos; "Pattern Recognition", 4th Edition, Academic Press, 2009, ISBN 978-1-59749-272-0
- Cristianini, Nello; and Shawe-Taylor, John; *An Introduction to Support Vector Machines and other kernel-based learning methods* (<https://web.archive.org/web/20180627015707/https://www.support-vector.net/>), Cambridge University Press, 2000. ISBN 0-521-78019-5 (SVM Book)
- Huang, Te-Ming; Kecman, Vojislav; and Kopriva, Ivica (2006); *Kernel Based Algorithms for Mining Huge Data Sets* (<http://learning-from-data.com>), in *Supervised, Semi-supervised, and Unsupervised Learning*, Springer-Verlag, Berlin, Heidelberg, 260 pp. 96 illus., Hardcover, ISBN 3-540-31681-7
- Kecman, Vojislav; *Learning and Soft Computing — Support Vector Machines, Neural Networks, Fuzzy Logic Systems* (<http://www.support-vector.ws>), The MIT Press, Cambridge, MA, 2001.
- Schölkopf, Bernhard; and Smola, Alexander J.; *Learning with Kernels*, MIT Press, Cambridge, MA, 2002. ISBN 0-262-19475-9
- Schölkopf, Bernhard; Burges, Christopher J. C.; and Smola, Alexander J. (editors); *Advances in Kernel Methods: Support Vector Learning* (<https://web.archive.org/web/20071017030144/http://kernel-machines.org/nips97/book.html>), MIT Press, Cambridge, MA, 1999. ISBN 0-262-19416-3.
- Shawe-Taylor, John; and Cristianini, Nello; *Kernel Methods for Pattern Analysis* (<http://www.kernel-methods.net>), Cambridge University Press, 2004. ISBN 0-521-81397-2 (Kernel Methods Book)
- Steinwart, Ingo; and Christmann, Andreas; *Support Vector Machines* (<https://web.archive.org/web/20120220084913/http://www.staff.uni-bayreuth.de/~btms01/svm.html>), Springer-Verlag, New York, 2008. ISBN 978-0-387-77241-7 (SVM Book)
- Tan, Peter Jing; and Dowe, David L. (<http://www.csse.monash.edu.au/~dld>) (2004); *MML Inference of Oblique Decision Trees* (<http://www.csse.monash.edu.au/~dld/David.Dowe.publications.html#TanDowe2004>), Lecture Notes in Artificial Intelligence (LNAI) 3339, Springer-Verlag, pp. 1082–1088 (<https://web.archive.org/web/20160806013805/http://www.csse.monash.edu.au/~dld/Publications/2004/Tan+DoweAI2004.pdf>). (This paper uses minimum message length (MML) and actually incorporates probabilistic support vector machines in the leaves of decision trees.)
- Vapnik, Vladimir N.; *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995. ISBN 0-387-98780-0
- Vapnik, Vladimir N.; and Kotz, Samuel; *Estimation of Dependences Based on Empirical Data*, Springer, 2006. ISBN 0-387-30865-2, 510 pages [this is a reprint of Vapnik's early book

describing philosophy behind SVM approach. The 2006 Appendix describes recent development].

- Fradkin, Dmitriy; and Muchnik, Ilya; *Support Vector Machines for Classification* (<https://web.archive.org/web/20161019164722/http://paul.rutgers.edu/~dfradkin/papers/svm.pdf>) in Abello, J.; and Carmode, G. (Eds); *Discrete Methods in Epidemiology*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, volume 70, pp. 13–20, 2006. Succinctly describes theoretical ideas behind SVM.
- Bennett, Kristin P.; and Campbell, Colin; *Support Vector Machines: Hype or Hallelujah?* ([http://kdd.org/exploration\\_files/bennett.pdf](http://kdd.org/exploration_files/bennett.pdf)), SIGKDD Explorations, 2, 2, 2000, 1–13. Excellent introduction to SVMs with helpful figures.
- Ivanciuc, Ovidiu; *Applications of Support Vector Machines in Chemistry* ([http://www.ivanciuc.org/Files/Reprint/Ivanciuc\\_SVM\\_CCR\\_2007\\_23\\_291.pdf](http://www.ivanciuc.org/Files/Reprint/Ivanciuc_SVM_CCR_2007_23_291.pdf)), in *Reviews in Computational Chemistry*, Volume 23, 2007, pp. 291–400.
- Catanzaro, Bryan; Sundaram, Narayanan; and Keutzer, Kurt; *Fast Support Vector Machine Training and Classification on Graphics Processors* (<https://web.archive.org/web/20120302180102/http://www.eecs.berkeley.edu/~catanzar/icml2008.pdf>), in *International Conference on Machine Learning*, 2008
- Campbell, Colin; and Ying, Yiming; *Learning with Support Vector Machines* (<http://www.morganclaypool.com/doi/abs/10.2200/S00324ED1V01Y201102AIM010>), 2011, Morgan and Claypool. ISBN 978-1-60845-616-1.
- Ben-Hur, Asa, Horn, David, Siegelmann, Hava, and Vapnik, Vladimir; "Support vector clustering" (2001) *Journal of Machine Learning Research*, 2: 125–137.

## 外部链接

---

- [www.support-vector.net](https://web.archive.org/web/20180627015707/https://www.support-vector.net/) (<https://web.archive.org/web/20180627015707/https://www.support-vector.net/>) The key book about the method, "An Introduction to Support Vector Machines" with online software
- Burges, Christopher J. C.; A Tutorial on Support Vector Machines for Pattern Recognition (<http://research.microsoft.com/en-us/um/people/cburges/papers/svmtutorial.pdf>), *Data Mining and Knowledge Discovery* 2:121–167, 1998
- [www.kernel-machines.org](http://www.kernel-machines.org) (<http://www.kernel-machines.org>) (*general information and collection of research papers*)
- [www.support-vector-machines.org](http://www.support-vector-machines.org) (<http://www.support-vector-machines.org>) (*Literature, Review, Software, Links related to Support Vector Machines — Academic Site*)
- [svmtutorial.online](https://svmtutorial.online) (<https://svmtutorial.online>) A simple introduction to SVM, easily accessible to anyone with basic background in mathematics
- [videlectures.net](http://videlectures.net/Top/Computer_Science/Machine_Learning/Kernel_Methods/Support_Vector_Machines/) ([http://videlectures.net/Top/Computer\\_Science/Machine\\_Learning/Kernel\\_Methods/Support\\_Vector\\_Machines/](http://videlectures.net/Top/Computer_Science/Machine_Learning/Kernel_Methods/Support_Vector_Machines/)) (*SVM-related video lectures*)
- Karatzoglou, Alexandros et al.; Support Vector Machines in R (<http://www.jstatsoft.org/v15/i09/paper>), *Journal of Statistical Software* April 2006, Volume 15, Issue 9.
- [libsvm](http://www.csie.ntu.edu.tw/~cjlin/libsvm/) (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>), LIBSVM is a popular library of SVM learners
- [liblinear](http://www.csie.ntu.edu.tw/~cjlin/liblinear/) (<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>) is a library for large linear classification including some SVMs
- Shark (<https://web.archive.org/web/20100227221243/http://shark-project.sourceforge.net/>) is a C++ machine learning library implementing various types of SVMs
- dlib (<http://dlib.net/ml.html>) is a C++ library for working with kernel methods and SVMs
- SVM light (<https://web.archive.org/web/20150419060655/http://svmlight.joachims.org/>) is a collection of software tools for learning and classification using SVM

- SVMJS live demo (<http://cs.stanford.edu/people/karpathy/svmjs/demo/>) is a GUI demo for JavaScript implementation of SVMs
  - Gesture Recognition Toolkit (<https://github.com/nickgillian/grt>) contains an easy to use wrapper for libsvm (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>)
- 

取自“<https://zh.wikipedia.org/w/index.php?title=支持向量机&oldid=61566642>”

---

本页面最后修订于2020年9月9日 (星期三) 13:02。

本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用。（请参阅使用条款）

Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。

维基媒体基金会是按美国国内稅收法501(c)(3)登记的非营利慈善机构。