# 支持向量机

支持向量机（Support Vector Machines, SVM）：是一种**二分类模型**，它的基本模型是定义在特征空间上的间隔最大化的线性分类器，**间隔最大化**使它有别于感知器；支持向量机还包括核技巧（通过非线性函数转换为线性模型），这使它成为实质上的非线性分类器。

支持向量机的学习策略就是**间隔最大化**，可以形式化为一个求解**凸二次规划**（convex quadratic programming）的问题，也等价于正则化的**合页损失函数最小化问题**，支持向量机的学习算法是求解凸二次规划的最优算法。[1]

按训练数据分线性可分程度，支持向量机的分类以及对应采取的算法如下：

1. 线性可分支持向量机：硬间隔最大化（线性可分）
2. 线性支持向量机：软间隔最大化（近似线性可分）
3. 非线性支持向量机：核技巧+软间隔最大化（线性不可分）

线性可分支持向量机可以看作是线性支持向量机的一个特例；非线性支持向量机则通过**核函数**将线性不可分数据转换为线性可分数据，从而转换为线性支持向量机。

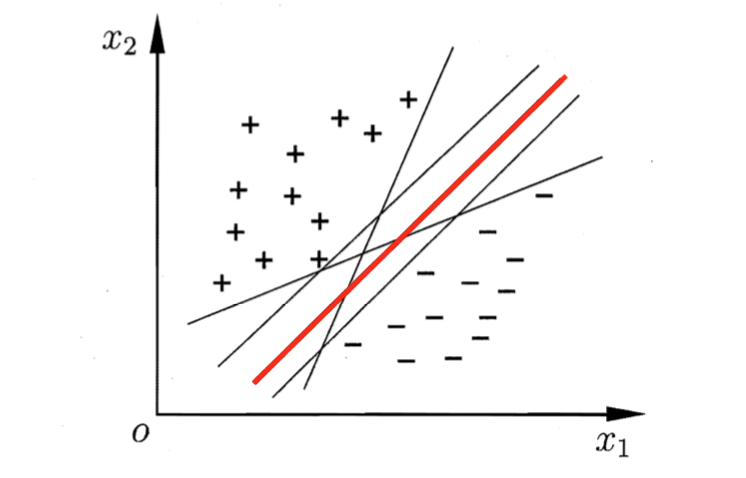
间隔最大化：按间隔中是否能出现样本点，分为硬间隔最大化和软间隔最大化。硬间隔最大化中不允许有样本点出现，因此适合于数据线性可分的情况；软间隔最大化中允许有少量样本点出现，适合于含有噪声的线性可分数据（称为近似线性可分）。

本文将按照上述思路介绍的三类支持向量机、核函数。

# 1. 线性可分支持向量机

**线性可分支持向量机**（linear support vector machine in linearly separable case）与**硬间隔最大化**（hard margin maximization）

首先我们来看一个线性可分的例子，给定训练样本，分类学习最基本的想法就是基于训练集 在样本空间中找到一个**划分超平面**，将不同类别的样本分开，但能将训练样本分开的划分超平面可能有很多，如下图所示：



直观上，应该去找位于两类训练样本“正中间”的划分超平面，因为该划分超平面对训练样本局部扰动的“容忍”性最好。例如，由于训练集的局限性或噪声的影响，训练集外的样本可能比图 1 中的训练样本更接近两个类的分隔届，这将使许多划分超平面出现错误，而红色的超平面受影响最小。换而言之，这个划分超平面所产生的分类结果鲁棒性是最好的，对预测数据的泛化能力也是最强的。

在样本空间中，**划分超平面**可通过如下线性方程来描述：

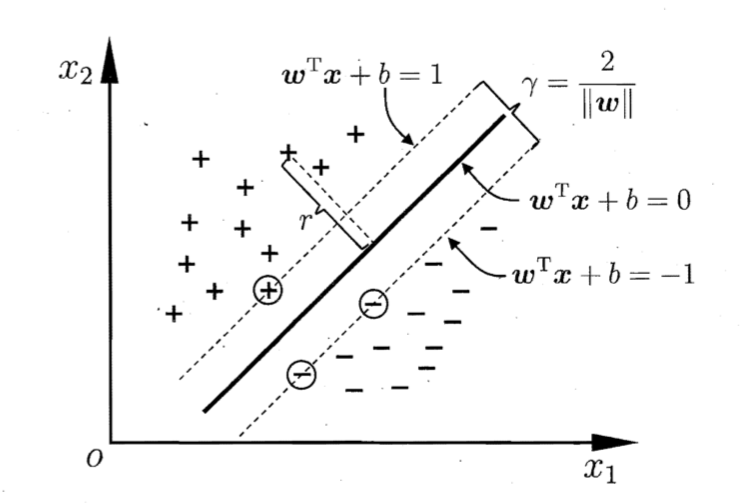
其中 为**法向量**，决定了超平面的方向； 为**位移项**，决定了超平面与原点之间的距离。因此划分超平面可以唯一的被法向量 和位移项 确定，我们将这个超平面记为 。

样本空间中任意点 到超平面 的距离可以写为：

假设超平面 能将训练样本正确分类，即对于 ，若 ，则有 ；若 ，则有 ，如下：

如图 2 所示，距离超平面**最近的**这几个训练样本点使公式 (3) 的等号成立，它们被称为“支持向量”（support vector），两个异类向量到超平面的距离之和为：

它被称为间隔（margin）。



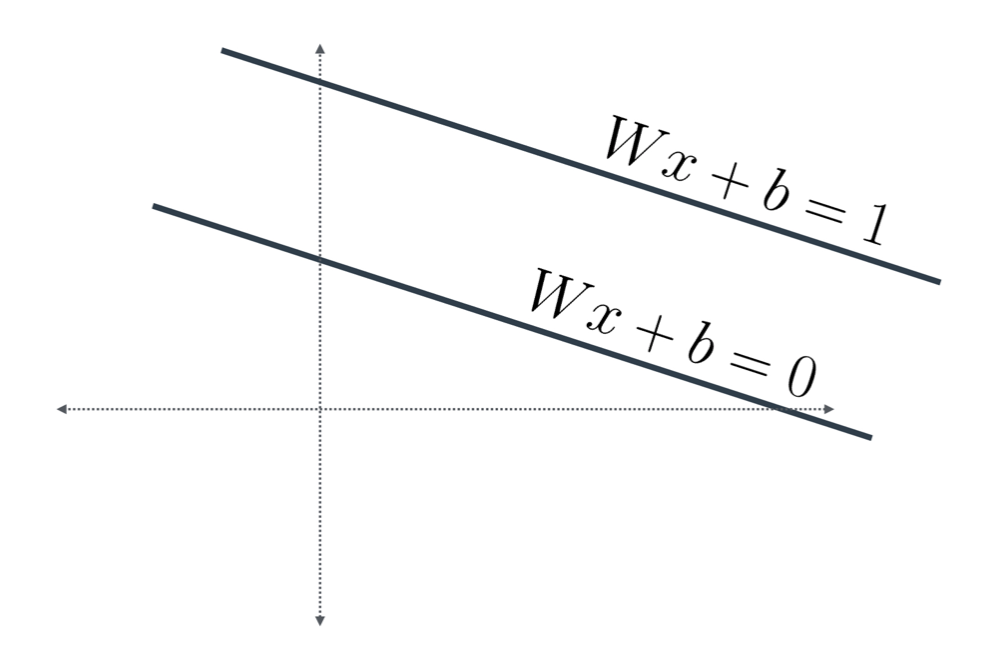
## 1.1 间隔计算公式推导

首先，设，并且 。

假设我们有三条直线：

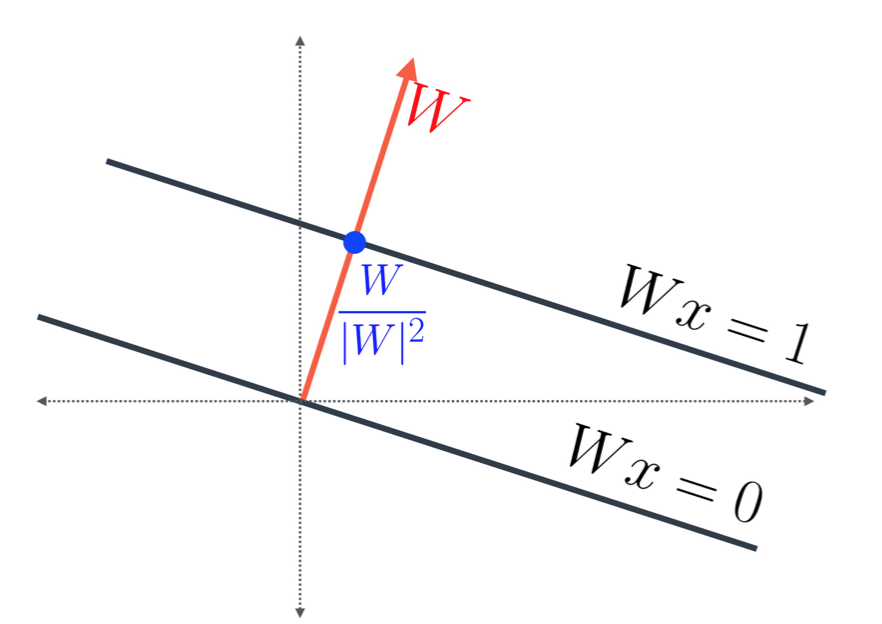
其中第二条为超平面的表达式，第一条和第三条为边界的表达式。由于这三条线为等距平行线，因此想要确定间隔（两条边界的距离），只需要计算边界与超平面之间的距离，并乘 2 就得到了间隔。

下面上边界到超平面的距离：



由于移动直线并不改边之间的距离，因此可以将超平面移动到与原点相交，此时直线方程变为：

现在，超平面的直线表达式为，因此其法向量为。

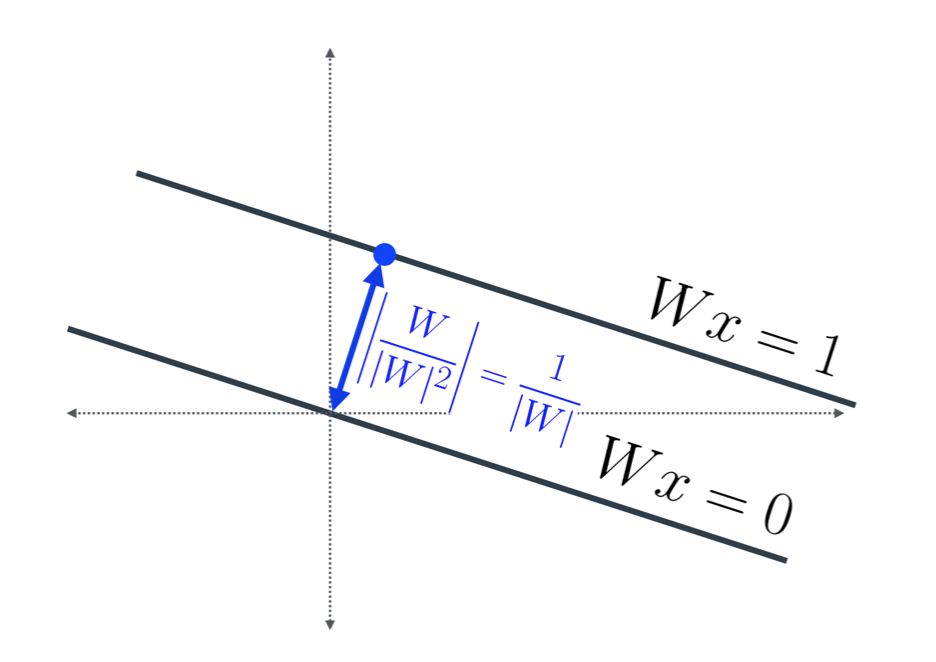


法向量 与边界相交于蓝点，假设该点的坐标为 ，由于该点位于法向量 上，所以 是 的倍数，即：

将点 代入边界方程 有：

因此，所以蓝点坐标可化为：

现在，两条直线之间的距离是蓝色向量的范数，由于分母是一个标量，向量 的范数正是 ，即 ，如下图所示：



最后，最终的间隔为两倍距离，即：

## 1.2 硬间隔最大化

支持向量机学习的基本想法是求解能够正确划分训练数据集并且间隔最大化的分离超平面。对于线性可分的训练数据集而言，线性可分的分离超平面有无数个，但是间隔最大的分离超平面是唯一的。

这里的间隔最大化又称为硬间隔最大化，间隔最大化的直观解释是：对训练数据集找到间隔最大的超平面以为着以充分大的确信度对训练数据进行分类，也就是说，不仅将正负实例点分开，而且最最难分的实例点（离超平面最近的点）也有足够大的确信度将它们分开。这样的超平面应该对未知的新实例有很好的分类预测能力。

求解硬间隔最大化下的超平面可以分为一下几步：

1. 将**间隔最大化**问题化为其**倒数的最小化**问题（为了应用拉格朗日对偶性求解最小化规划问题的解），我们将这个最小化问题称为**原始问题**，相应的算法1.1称为**最大间隔法**；
2. 利用**拉格朗日对偶性**将原始问题转换为拉格朗日函数，分两步求解得到算法1.1的最优解，我们将这个算法称为**对偶算法**；最后利用最优解代入公式可以得到**分离超平面**以及**分离决策函数**的方程式。

下面我将详细的介绍这两个步骤。

### 1.2.1 原始问题

由1.1可知，间隔的计算公式为，因此最大化间隔 相当于最小化 ，于是就得到下面的线性可分支持向量学习的最优化问题：

**算法1.1 线性可分支持向量机学习方法——最大间隔法**

输入：线性可分训练数据集，其中，，；

输出：最大间隔分离超平面和分类决策函数。   
（1） 构造并求解约束时最优化问题：

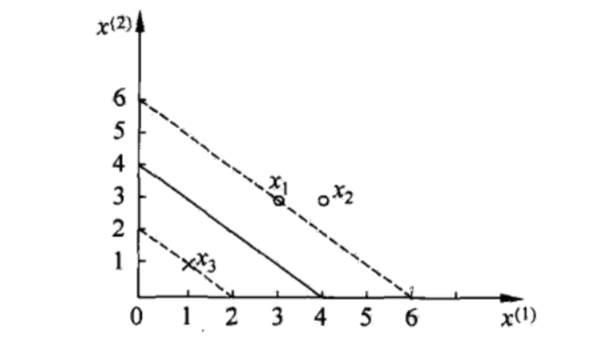
求得最优解。

（2）由此得到分离超平面：

分离决策函数：

**例 1.1**

已知一个如图 6 所示的训练数据集，其正例点是 ，负例点是 ，试求最大间隔分离超平面。



按照算法1.1，根据训练数据集构造约束最优化问题：

求得此最优化问题的解 ，于是最大间隔分离超平面为：

其中，与 为支持向量。

### 1.2.2 对偶算法

为了求得算法1.1 的最优化约束条件下的解 ，我们将它作为原始最优化问题，利用**拉格朗日对偶性**，通过求对偶问题（dual problem）得到原始问题（primal problem）的最优解，这就是线性可分支持向量机的**对偶算法**。

首先建立拉格朗日函数（Lagrange function），为此，对不等式约束条件

引入拉格朗日乘子（Largrange multiplier）:

定义拉格朗日函数：

其中， 为拉格朗日乘子向量。

根据**拉格朗日对偶性**，原始问题的对偶问题是极大极小问题：

所以为了得到对偶问题的解，需要**先对 求 的极小，再对 求极大。**

(1) 求

将拉格朗日函数 分别对 求偏导数，并令其等于0:

得：

将 (13) 代入拉格朗日公式 (10)，得：

即

（2）求 对 的极大，即使对偶问题：

将公式 (16) 的最大化转换为最小化，就得到了下面与之等价的对偶最优化问题：

设 是对偶最优化问题 (18-19) 的解，则存在下标 ，并可按下式求得原始最优化问题 (5) 的解 ：

**证明** 对于原始问题和对偶问题，KKT（Karush-Kuhu-Tucker）条件成立，即得：

由此得：

其中至少有一个 （反证法，假设 ，由 (22) 第一条公式可知 ，而 不是原始最优化问题 (5) 的解，产生矛盾），对此 有：

将 (20) 代入 (24)，并注意到 ，即得：

将 代入公式 (6)、(7)可得分离超平面的计算公式：

分类决策函数可以写为：

这就是说，对于给定的线性可分数据集，可以首先求对偶问题 (18-19) 的解 ；再利用公式 (20-21) 求得原始问题的解 ；从而根据公式 (6-7) 得到分离超平面以及分类决策函数。

这种算法分称为**线性可分支持向量机的对偶学习算法**，是线性可分支持向量机学习的基本算法。

我们可以将上述过程总结为算法 1.2。

**算法1.2 线性可分支持向量机学习算法**

输入：线性可分训练数据集，其中，，；

输出：最大间隔分离超平面和分类决策函数。   
（1） 构造并求解约束时最优化问题：

求得最优解 。

（2）计算

$w^\*=\sum\_{i=1}^N \alpha\_i^\*y\_ix\_i\tag{30}$

并选择 的一个正分量 ，计算：

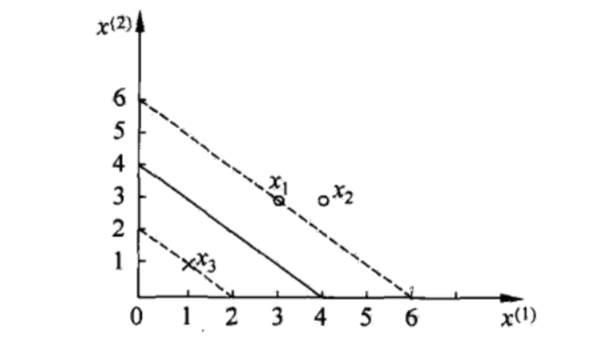
$b^\*=y\_j-\sum\_{i=1}^Na\_i^\*y\_i(x\_i\cdot x\_j)\tag{31}$

（3）求得分离超平面：

分离决策函数：

**例 1.2**

与例1.1 相同，对一个如图 7 所示的训练数据集，其正例点是 ，负例点是 ，试用算法1.2求最大间隔分离超平面。



根据所给数据，对偶问题是：

解着一最优化问题，将 代入目标函数并记为：

对 求偏导数并令其为 0，易知 在点 取极值，但该点不满足约束条件 ，所以最小值应在边界上达到。

当 时，最小值 ；当 时，最小值 ，于是 在 达到最小，此时 。

由于 ，所以由1.3可知其对应的实例点 是支持向量。根据公式 (30-31) 计算得：

分离超平面为：

分类决策函数为：

## 1.3 支持向量

考虑原始最优化问题 (5) 以及对偶最优化问题 (28-29)，将训练数据集中对应与 的样本点 的实例 称为支持向量。

根据这一定义，支持向量一定在间隔边界上，由 KKT 互补条件可知：

对应与 的实例 ，有：

或

即 一定在间隔边界上，这里的支持向量的定义与之前给出的支持向量的定义是一致的。

# 2. 线性支持向量机

线性可分问题的支持向量机学习方法，对线性不可分训练数据是不使用的，因为这是上述方法中的不等式约束并不能都成立。因此此时需要修改硬间隔最大化，使其称为软间隔最大化。

## 2.1 软间隔最大化

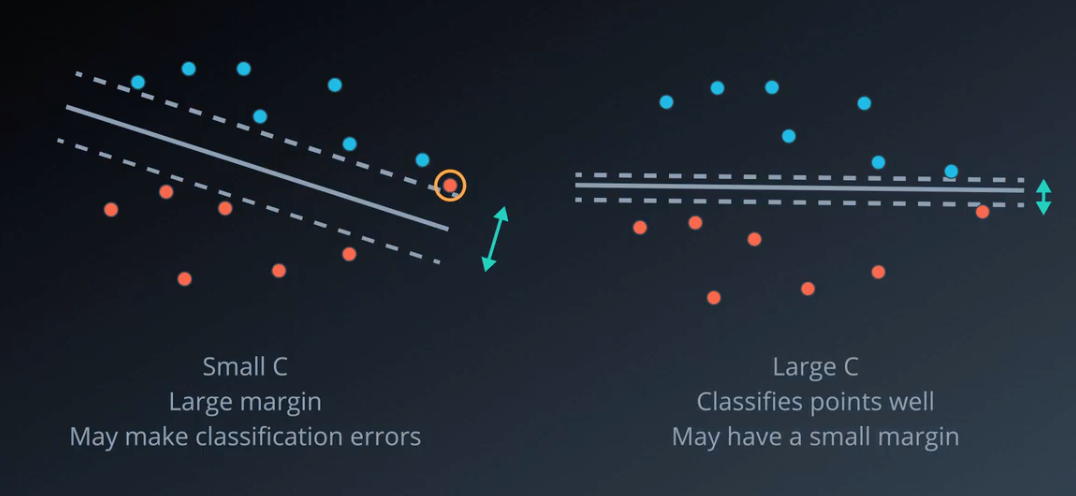
假定给定一个特征空间上的训练数据集：

其中， 为第 个特征向量， 为 的类标记。再假设训练数据集不是线性可分的。通常情况是，训练数据中有一些特异点（outlier），将这些特异点去除后，剩下大部分的样本点组成的集合是线性可分的，我们称之为近似线性可分的。

近似线性可分意味着某些样本点 不能满足函数间隔大于 1 的约束条件 (5)。为了解决这个问题，可以对每个样本点 引入一个松弛变量 ，使函数间隔加上松弛变量大于等于 1 。这样，约束条件变为：

同时，对每个松弛变量 ，支付一个代价 ，目标函数由原来的 变成：

这里， 称为惩罚参数，一般由应用问题决定， 值越大时对误分类的惩罚增大， 值小时对误分类的惩罚减小，如下图所示：



最小化目标函数 (38) 有两层含义：使 尽量小即间隔尽量大，同时使误分类点的个数尽量小， 是调和二者的系数。

有了上面的思路，可以和训练数据集线性可分时一样来考虑训练数据集线性近似可分时的线性支持向量机学习问题，相应于硬间隔最大化，它称为软间隔最大化（soft margin maximization）。

### 2.1.1 原始问题

近似线性可分的线性支持向量机的学习问题变成如下凸二次规划问题（**原始问题**）：

原始问题 (39-40) 是一个凸二次规划问题，因而关于 的解时存在的。可以证明 的解时唯一的， 但 的解不唯一，但可以证明 的解存在一个区间。

### 2.1.2 对偶算法

原始最优化问题 (39-40) 的拉格朗日函数是：

$\begin{equation} L(w, b, \xi, \alpha, \mu) \equiv \frac{1}{2}\|w\|^{2}+C \sum\_{i=1}^{N} \xi\_{i}-\sum\_{i=1}^{N} \alpha\_{i}\left(y\_{i}\left(w \cdot x\_{i}+b\right)-1+\xi\_{i}\right)-\sum\_{i=1}^{N} \mu\_{i} \xi\_{i}\tag{41} \end{equation}$

其中，。

与 1.2.2 相同，对偶问题是拉格朗日函数的极大极小问题，首先求 对 的极小并代入 (41) 得：

$\begin{equation} \min \_{w, b, \xi} L(w, b, \xi, \alpha, \mu)=-\frac{1}{2} \sum\_{i=1}^{N} \sum\_{j=1}^{N} \alpha\_{i} \alpha\_{j} y\_{i} y\_{j}\left(x\_{i} \cdot x\_{j}\right)+\sum\_{i=1}^{N} \alpha\_{i}\tag{42} \end{equation}$

再对 求 的极大，即得对偶问题：

$\begin{equation} \quad\quad\quad\quad\quad\max \limits\_{\alpha}\bigg[-\frac{1}{2} \sum\_{i=1}^{N} \sum\_{j=1}^{N} \alpha\_{i} \alpha\_{j} y\_{i} y\_{j}\left(x\_{i} \cdot x\_{j}\right)+\sum\_{i=1}^{N} \alpha\_{i}\bigg]\tag{43} \end{equation}$

$\begin{equation} \begin{array}{ll}{\text { s.t. }} & {\sum\_{i=1}^{N} \alpha\_{i} y\_{i}=0} \\ {} & {C-\alpha\_{i}-\mu\_{i}=0} \\ {} & {\alpha\_{i} \geqslant 0} \\ {} & {\mu\_{i} \geqslant 0, \quad i=1,2, \cdots, N}\end{array}\tag{44} \end{equation}$

将对偶最优化问题 (43-44) 进行变换：利用 (44) 第二条公式消去 ，从而只留下变量 ，同时将目标函数求极大转换为求极小，可得原始问题 (39-40)的**对偶问题** (45-46)：

$\begin{equation} \quad\quad\quad\quad\quad\quad\quad\quad\min \limits\_{\alpha}\bigg[\frac{1}{2} \sum\_{i=1}^{N} \sum\_{j=1}^{N} \alpha\_{i} \alpha\_{j} y\_{i} y\_{j}\left(x\_{i} \cdot x\_{j}\right)-\sum\_{i=1}^{N} \alpha\_{i}\bigg]\tag{45} \end{equation}$

$\begin{equation} \begin{array}{ll}{\text { s.t. }} & {\sum\_{i=1}^{N} \alpha\_{i} y\_{i}=0} \\ {} & {0 \leq \alpha\_i \leq C} \\ \end{array}\tag{46} \end{equation}$

设 是对偶问题 (45-46) 的一个解，若存在 的一个分量 ，，则原始问题 (39-40) 的解 可按下式求得：

$w^\*=\sum\_{i=1}^N\alpha\_i^\*y\_ix\_i\tag{47}$

$b^\*=y\_i-\sum\_{i=1}^Ny\_i\alpha\_i^\*(x\_i\cdot x\_i)\tag{48}$

综上，可得线性支持向量机学习算法。

**算法2 线性支持向量机学习算法**

输入：线性可分训练数据集，其中，，；

输出：分离超平面和分类决策函数。   
（1） 选择惩罚参数 ，构造并求解约束时最优化问题：

$\begin{equation} \quad\quad\quad\quad\quad\quad\quad\quad\min \limits\_{\alpha}\bigg[\frac{1}{2} \sum\_{i=1}^{N} \sum\_{j=1}^{N} \alpha\_{i} \alpha\_{j} y\_{i} y\_{j}\left(x\_{i} \cdot x\_{j}\right)-\sum\_{i=1}^{N} \alpha\_{i}\bigg]\tag{49} \end{equation}$

$\begin{equation} \begin{array}{ll}{\text { s.t. }} & {\sum\_{i=1}^{N} \alpha\_{i} y\_{i}=0} \\ {} & {0 \leq \alpha\_i \leq C} \\ \end{array}\tag{50} \end{equation}$

求得最优解 。

（2）计算

$w^\*=\sum\_{i=1}^N\alpha\_i^\*y\_ix\_i\tag{51}$

并选择 的一个分量 ，计算：

$b^\*=y\_i-\sum\_{i=1}^Ny\_i\alpha\_i^\*(x\_i\cdot x\_i)\tag{52}$

（3）求得分离超平面：

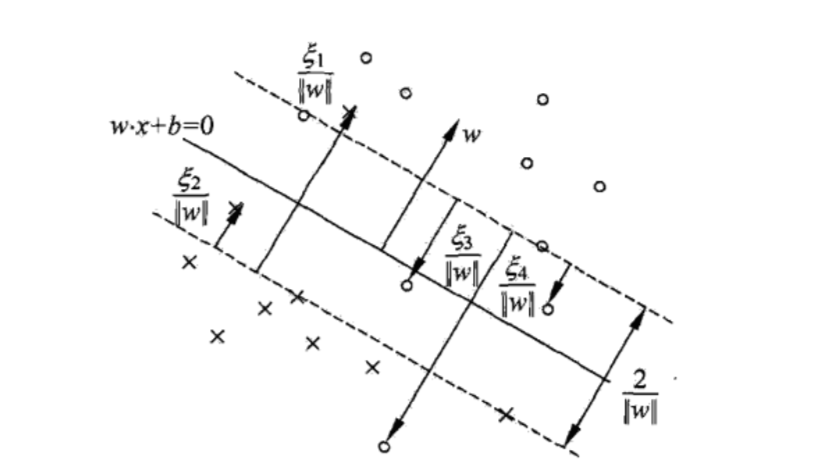
分离决策函数：

步骤 (2) 中，对任一适合条件 的 ，按 (52) 都可以计算出 ，但是由于原始问题 (43-44) 对 的解并不唯一，所以实际计算时可以取所有符合条件的样本点上的平均值。

## 2.2 支持向量

在线性不可分的情况下，将对偶问题 (49-50) 的解 中对应于 的样本点 的实例 称为支持向量（软间隔的支持向量）。

如图 9 所示，分离超平面由实现表示，间隔边界由虚线表示，正例点由“”表示，负例点由“”表示。实例 到间隔边界的距离为 。



软间隔的支持向量 或者在间隔边界上，或者在间隔边界与分离超平面之间，或者在分离超平面误分一侧：

* 若 ，则 ，支持向量 恰好落在区间边界上；
* 若 ， ，则分类正确，支持向量 在区间边界与分离超平面之间；
* 若 ， ，则支持向量 在分离超平面上；
* 若 ， ，则支持向量 在分离超平面误分一侧。

## 2.3 合页损失函数

对于线性支持向量机来说，其模型为分离超平面 以及决策函数 ，其学习策略为软间隔最大化，学习算法为凸二次规划。

线性支持向量机学习还有另外一种解释，就是最小以下目标函数：

$\sum\_{i=1}^N[1-y\_i(w \cdot x\_i+b)]\_+ + \lambda \|w\|^2 \tag{55}$

1. 目标函数的第一项式经验损失或经验风险。

函数：

$L(y(w \cdot x+b)) = [1-y(w \cdot x+b)]\_+ \tag{56}$

称为合页损失函数（hinge loss function）。下标“+”表示以下取正值的函数：

\begin{equation} [z]\_{+}=\left{\begin{array}{ll}{z,} & {z>0} \ {0,} & {z \leqslant 0}\end{array}\right.\tag{57} \end{equation}

这就是说，当样本点 被正确分类且函数间隔（确信度） 大于 1 时，损失时0，否则损失是 ，注意图 9 中的实例点 被正确分类但损失不是 0 。

1. 目标函数的第 2 项是系数为 的 的 范数，是正则化项。

**定理**

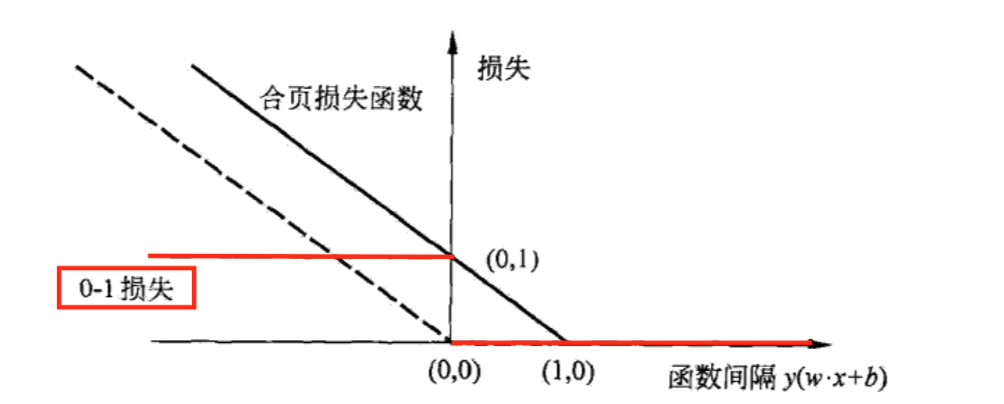
线性支持向量机原始最优化问题：

$\min \limits\_{w,b,\xi}\quad \frac{1}{2}\|w\|^2+C\sum\_{i=1}^N\xi\_i\tag{58}$

等价于最优化问题：

$\min \limits\_{w,b}\quad \sum\_{i=1}^N[1-y\_i(w \cdot x\_i+b)]\_+ + \lambda \|w\|^2 \tag{60}$

合页损失函数的图形如下图所示，横轴是函数间隔 ， 纵轴是损失，由于函数形状像一个合页，故命名合页损失函数。



图中还画出 0-1 损失函数，可以认为它是二类分类问题的真正的损失函数，而合页损失函数是 0-1 损失函数的上界，由于 0-1 损失函数不是连续可导的，直接优化由其构成的目标函数比较困难，可以认为线性支持向量机就是优化由 0-1 损失函数的上界（合页损失函数）构成的目标函数。这时的上界损失函数由称为代理损失函数（surrograte loss function）。

图中虚线显示的是感知器的损失函数 。这时，当样本点 被正确分类时，损失是 0，否则损失是 ，相比之下，合页损失函数不仅要分类正确，而且确信度足够高时损失才是 0，也就是说，合页损失函数对学习由更高的要求。

# 参考资料

[1] 李航. 统计学习方法[M]. 北京: 清华大学出版社, 2012: 95-115.

[2] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016: 121-126.