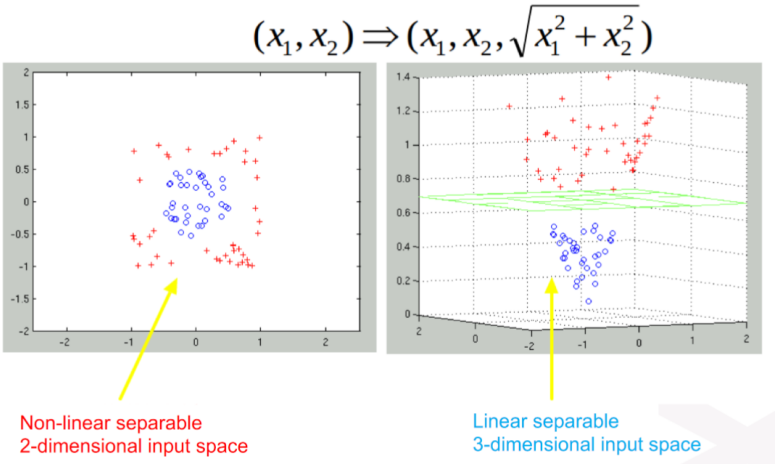
INT104W6_支持向量机（SVM）

引入线性分类

我们有一组红色和蓝色的点，要想把它们按照颜色分成两份，该怎么分？



在二维空间中，我们需要围着蓝色点画一个圈，恰能将整个特征空间分为红蓝两份，但要进行这样的**非线性的分割是不容易**的，谁知道我们该用什么样的曲线去分割平面上属于不同类别的样本？

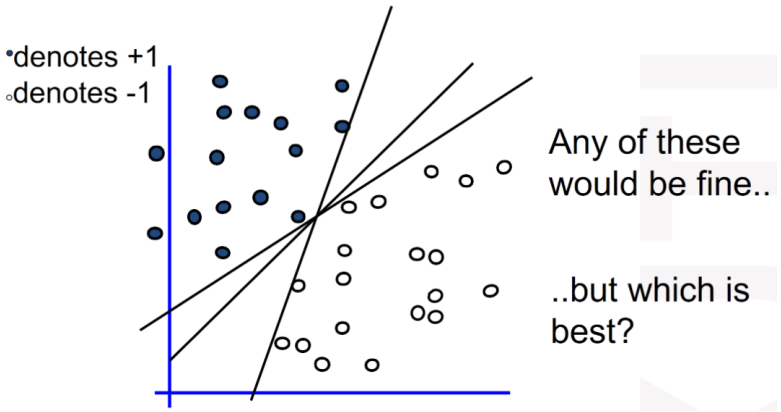
但如果我们把维度从二维投射至三维（增加的维度可能是现有特征的幂或组合，从而捕获特征和目标变量之间的非线性关系），就能很容易地**用线性的方式**（超平面）将新的特征空间一刀两断，一侧是红，一侧是蓝，轻易分开不同label的样本。这样的空间也就具有**线性可分性（linear separability）**。

将这种手段延拓到多维的超空间中，我们就能解决在低维空间中难以区分的样本。这就是支持向量机的主要想法。

支持向量机（SVM）是通过监督学习方式在数据集所在的空间建立**超平面（hyperplane）**划分数据集，从而进行二元分类的广义线性分类器（generalized linear classifier）。

现在平面上有一组可线性可分的样本，如何用SVM将它们良好地分类呢？

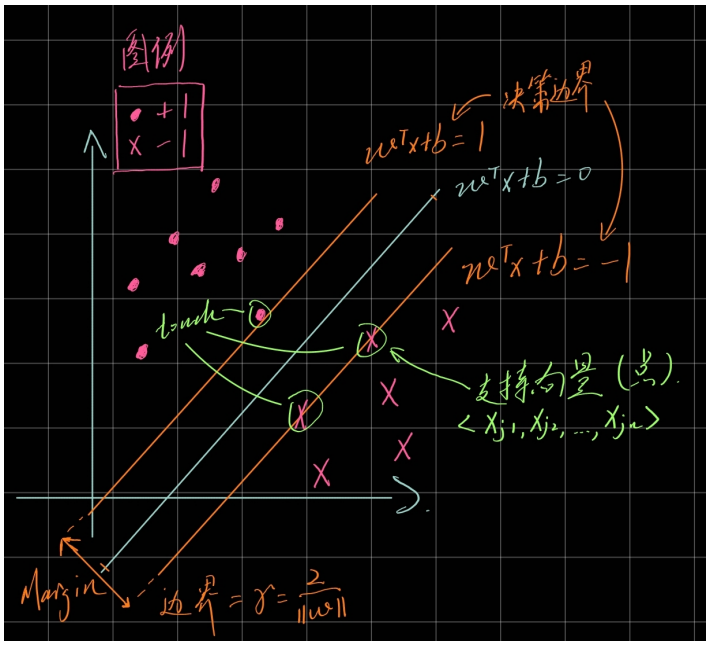
线性(Linear)支持向量机（LSVM）



线性分类器

我们需要找到一个**超平面（在n维的样本空间中维度为n-1的线性子空间，在这里是二维空间中一维的线）**来最有效地将空间分成正例与负例的两个半空间。

为使得分割最有效，我们需要确保这个分割可以最大限度地容忍噪声与新的**样本（instances）**，即**泛化能力最强。**



一种朴素的想法是在两类样本间设置一个分割（图中青色直线），使这根直线离两类样本的距离最大。可以想象，如果这根直线的宽度（**margin**，即**决策边界{decision boundary)的间距**）向两侧不断扩张，直到能同时接触到两类不同实例（instances）且宽度最大。此时直线所在的位置就是能最有效分割两类实例的**超平面（hyperplane）**；距离超平面间隔最小的点（实例）就是**支持向量**（Support Vector，位于决策边界上/附近，**支撑**超平面位置的点（向量））。

支持向量机的意思就是使超平面和支持向量之间的间隔尽可能的大，这样才可以使两类样本准确地分开。更大的Margin意味着分类器对样本点的分类更加确信，置信度更高，因此其泛化能力也更强，不易过拟合。**我们的目标就是要让margin尽可能大!**

由上我们也可以发现，决定超平面位置的并不是所有的样本点，而是其中少数的支持向量，其他的样本点（即非支持向量）对决策边界的位置没有影响。这意味着即使移除了这些非支持向量，决策边界也不会发生改变。因此，SVM只需要通过这些支持向量来决策，剩余的样本点则不参与。通过少数关键样本点（即支持向量）来确定决策边界，实现分类任务，使得SVM在小样本、高维和非线性分类问题中表现出色。

不可不知的中学几何公式

假设平面上有一条直线：

$$\text{直线一般式：} ax + by + c = 0 \tag{1}$$
$$\text{可记为：} w_1 x_{j1} + w_2 x_{j2} + b = 0 \tag{2}$$

点到直线距离

平面上又有一点$X_1(x_{11}, x_{12})$，求点到直线距离？

我们都学过点到直线距离公式，代入即可：

$$d = \frac{|w_1 x_{11} + w_2 x_{12} + b|}{\sqrt{w_1^2 + w_2^2}}$$

这一结论容易延拓到高维，对向量 $x1(x_{11}, x_{12}, ..., x_{1n})$ 到超平面 $w^Tx+b=0$ (其中 $w(w_1, w_2, ...,w_n)$)的距离：

$$d = \frac{|w_1 x_{11} + w_2 x_{12} + ... + w_n x_{1n} + b|}{\sqrt{w_1^2 + w_2^2 + ... + w_n^2}} \tag{3}$$
$$= \frac{|w^T x + b|}{||w||} \tag{4}$$

点在不同半空间（分割得到）

回到二维情况，如果我们定义在斜线一侧高于直线的点的距离是正数，另一侧低于直线的点的距离是负数以区分处于被直线分割不同的点，该怎么区分两侧的点？此时去掉绝对值即可：

$$判断位置 = \frac{w_1x_{11} + w_2x_{12} + b}{\sqrt{w_1^2 + w_2^2}} \propto w_1x_{11} + w_2x_{12} + b \tag{5}$$

$$\begin{cases} w_1x_{11} + w_2x_{12} + b \geq 0, 点重合或高于直线 \\ w_1x_{11} + w_2x_{12} + b < 0, 点低于直线 \end{cases} \tag{6}$$

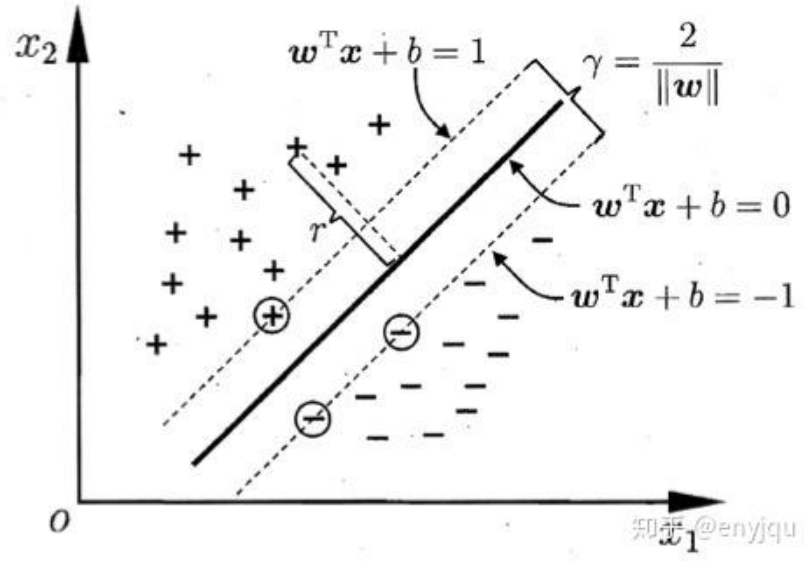
$$延拓至高维：\begin{cases} w^Tx + b \geq 0, 点重合或高于直线 \\ w^Tx + b < 0, 点低于直线 \end{cases} \tag{7}$$

如果我们为训练集的每个样本加上类别标签y(y∈(-1, 1))以表明该点应当属于的类别：将高于分类超平面的样本类别设定为1，低于的设为-1；那么在训练集X{(x1,y1), (x2,y2), ..., (xn,yn)}中我们发现：（低于超平面的样本负负得正）

$$对所有的样本X_i，如果(w,b)满足：y_i(w^Tx_i + b) \geq 0 \tag{8}$$

这样的分割就是能将空间分成正例与负例的两个半空间的，这样的样本空间也是线性可分的。

但是这个条件在实际应用中往往难以达成，并不是所有样本一高于超平面就是正例，一低于就属于负例。我们给定上下两个决策边界（两个支持超平面），如果你高于超平面足够的距离，那我才足够确信你是正例，反之亦然。



$$\begin{cases} w^Tx + b \geq 1, 高于决策上界 \\ w^Tx + b < 1, 低于决策下界 \end{cases} \tag{9}$$

$$\Rightarrow 对X_i，若(w,b)满足：y_i(w^Tx_i + b) \geq 1，则足够可分 \tag{10}$$

优化问题：具有线性约束的凸二次优化

在上面的推导中，我们得到了：**向量到超平面距离（点到直线距离）**，**确保线性可分（点在不同半空间）**两个计算方法

实际上这就是计算SVM所需要的：**在确保线性可分的情况下 使支持向量到超平面距离 尽可能大。**

我们需要用一些小技巧**把计算SVM转化为解具有线性约束（线性可分）的凸二次优化（距离最大）问题**（convex quadratic optimization problems with linear constraints）

$$\text{对超平面：} w^Tx+b=0 \tag{11}$$

$$\text{等效于：} \alpha\cdot w^Tx+\alpha\cdot b=0 \quad (\alpha\in\mathbb{R}^+) \tag{12}$$

$$\Rightarrow \alpha\cdot(w^Tx+b)=0 \tag{13}$$

$$\text{由}\underline{\textbf{式(8)}}\text{，如果}(\alpha w,\alpha b)\text{满足：}\alpha\cdot y_i(w^Tx_i+b)\geq 0 \quad \text{则此分割线性可分} \tag{14}$$

$$\text{总存在一个缩放值}\alpha\text{,使：}\alpha\cdot y_i(w^Tx_i+b)=1 \tag{15}$$

$$\text{由}\underline{\textbf{式(4)}}: d=\frac{|w^Tx+b|}{||w||}\Rightarrow d=\frac{1}{||w||} \tag{16}$$

这里求出了一侧的支持向量到超平面的距离d，由于两侧距离相同，我们也能求出两个异类支持向量到超平面的**距离之和**，也称为**间隔（margin）**：

$$\text{间隔}(margin)\text{：}\gamma=2d=\frac{2}{||w||} \tag{17}$$

$$\text{为使间隔最大，要让}w\text{的}L2\text{范数最小：}\gamma\uparrow=\frac{2}{||w||\downarrow} \tag{18}$$

$$\text{综上，由}\underline{\textbf{式(10)}}\text{:} \tag{19}$$

$$\textbf{即找到}\quad min_{w,b}\frac{1}{2}||w||^2 \tag{20}$$

$$\textbf{能满足}\quad y_i(w^Tx_i+b)\geq 1 \quad i\in(1,2,...,n) \tag{21}$$

这是一个解不等式约束的凸优化问题，我们使用拉格朗日乘子法、对偶问题、互补松弛性、KKT条件，最终能解出最优超平面。发现决定最佳超平面时只有支持向量起作用，而其他数据点并不起作用。

优点：

- 由于SVM是一个凸优化问题，所以求得的解一定是全局最优而不是局部最优。
- 不仅适用于线性线性问题，还适用于非线性问题（核）。

- 拥有高维样本空间的数据也能用SVM，这是因为数据集的复杂度只取决于支持向量而不是数据集的维度，这在某种意义上避免了“维数灾难”。
- 理论基础比较完善 （如神经网络就更像黑盒子）。

缺点：

- 二次规划问题求解将涉及m阶矩阵的计算（m为样本的个数），因此SVM不适用于超大数据。（SMO算法可以缓解这个问题。）
- 只适用于二分类问题。（SVM的推广SVR也适用于回归问题；可以通过多个SVM的组合来解决多分类问题。）

使用注意：

SVM对数据规模敏感，特征参数的规模不同可能导致糟糕的决策边界，建议**预先使用sklearn的** `StandardScaler` **进行归一化。**

硬间隔分类（Hard margin Classification）

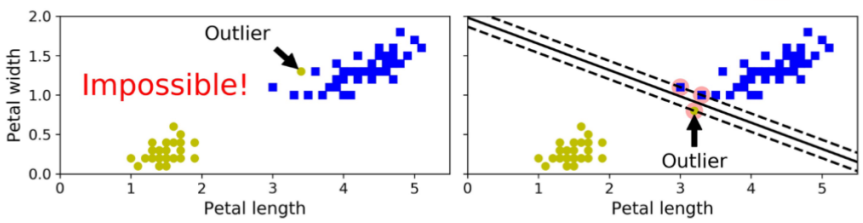


Figure 5-3. Hard margin sensitivity to outliers

如果所有相同类别的样本都被超平面分割在一侧，这就是硬间隔分类。但在很多情况下是不可能做到的（左图，原特征空间线性不可分或只是近似线性可分），而且很可能对异常值（outlier）或噪声敏感（右图，此分割泛化性很差）。

我们应当允许一些异常值的越界（margin violation）以确保对整体分类的性能。

软间隔分类（Soft margin Classification）

硬间隔： 找到 $min_{w,b}\frac{1}{2}||w||^2$ 能满足 $y_i(w^Tx_i+b)\geq 1 \quad i\in(1,2,...,n)$ (25)

允许部分越界： 找到 $min_{w,b}\frac{1}{2}||w||^2+C\cdot HingeLoss$ (26)

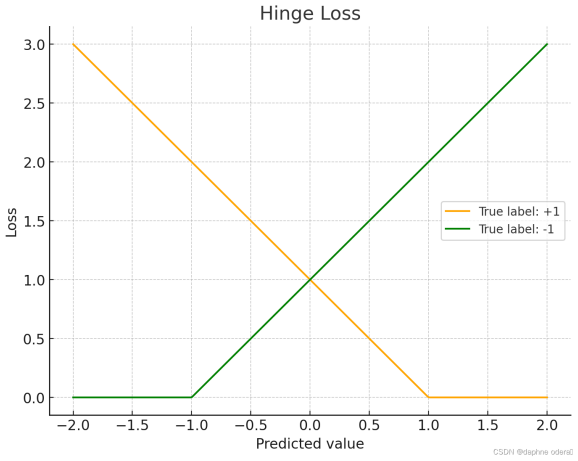
合页损失：$HingeLoss=\frac{1}{n}\sum_{i=1}^{n}max(0,1-y_i(w^Tx_i+b))$ (22)

$=\frac{1}{n}\sum_{i=1}^{n}max(0,1-y_i\hat{y}_i)$ (23) (27)

$=\frac{1}{n}\sum_{i=1}^{n}\xi_i$ (24)

软间隔： 找到 $min_{w,b}\frac{1}{2}||w||^2+C\cdot\sum_{i=1}^{n}\xi_i$ (28)

能满足 $y_i(w^Tx_i+b)\geq 1-\xi_i \quad i\in(1,2,...,n), \quad \xi_i\geq 0$ (29)



其中n是样本数量，y_i是真实标签(1, -1)，$\hat{y}_i$是模型的预测分数∈R，$C\cdot\sum_{i=1}^{n}\xi_i$是**松弛变量**（Slack variable），C是控制软硬度的超参数（hyper-parameter）。

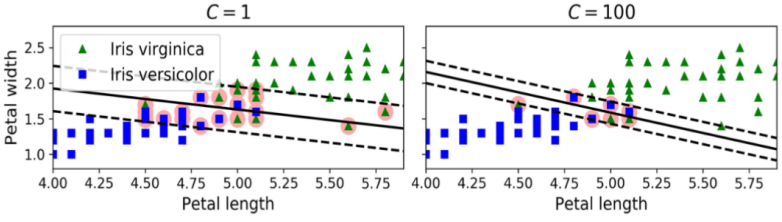


Figure 5-4. Large margin (left) versus fewer margin violations (right)

软间隔允许一定的样本越界，数量取决于超参数C，C越小允许越多样本越界，C越大则越严格。C 与正则化超参数α相反，要正则化SVM，需降低C。

代码实现：

线性SVC（LinearSVC）

```
LinearSVC(loss="hinge", C=1)
```

接受两个损失函数： “hinge” 和“squared_hinge”，默认为后者。

不会输出支持向量（请用 .intercept_ 与 .coef_ 找到训练集中的支持向量）。

请先使用StandardScaler中心化

如果训练实例数大于特征数，请设定 dual=False

SVC（Support Vector Classification）

```
SVC(kernel="linear", C=1)
```

如果是线性分类器请使用kernel= “linear”

如果需要硬间隔分类，请设置 C=float(“inf”) 或 C=1e10（一个大数）

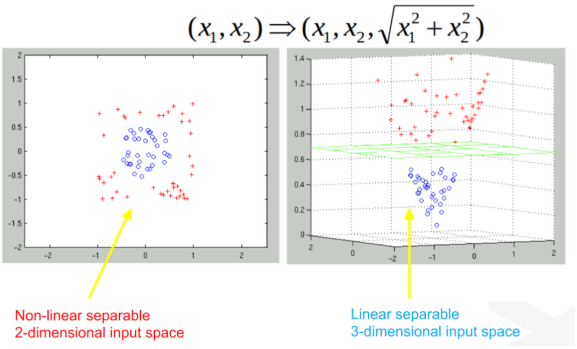
随机梯度下降分类器（SGD Classifier）

```
SGDClassifier(loss="hinge", alpha = 1/(m*C))
```

收敛速度慢，但适用于大数据集或在线数据集。

非线性(Nonlinear)支持向量机（使用kernel SVM）

回到最开始的问题，这实际上是一个非线性分类问题，因为原空间是线性不可分的，我们使用**核技巧**（kernel trick）增加了一个维度来区分这些样本。使用**核函数**（kernel function），将非线性问题转化为线性问题。非线性SVM使用核函数升维，来将数据映射到高维空间，将非线性问题转化为线性问题，从而在高维空间中找到一个线性分割的超平面来解决原来的非线性问题。



根据拉格朗日对偶问题，理论上我们可以通过求每个样本点的内积来求得最优的α向量，在样本较少时可行，但在N个样本点，M个特征的情况下，需要计算的内积数量为N!，总计算约M(M-1)N!次。在线性不可分的情况下，如果进行原空间到特征空间的映射，特征数量即维度将极大升高甚至是无穷维，这是不可接受的。

内积（点积）（Inner product）

$$对n维向量a,b，内积：\ \langle a,b\rangle = a\cdot b = \sum_{i=1}^{n} a_i b_i = a_1b_1 + a_2b_2 + ... + a_nb_n$$

此时我们需要使用核技巧，利用核函数升维，将非线性问题转化为线性问题，绕过暴力计算样本点内积的过程，而直接求得最终的内积。

主旨：把低维非线性空间转换为高维线性空间。

$$对n维空间x：\ x = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix} \qquad K(a,b) = a^T b = \begin{bmatrix} a_1 & a_2 & ... & a_n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ ... \\ b_n \end{bmatrix}$$

$$高维特征空间N >> n：\ \phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ ... \\ \phi_N(x) \end{bmatrix} \qquad K(\phi(a),\phi(b)) = \phi(a)^T\phi(b) \Rightarrow KernelFunction(a^Tb)$$

$$\textbf{kernel SVM：}\ 找到 \quad min_{w,b}\frac{1}{2}||w||^2 + C\cdot\sum_{i=1}^{n}\xi_i \tag{30}$$

$$能满足 \quad y_i(w^T\phi(x)_i + b) \geq 1 - \xi_i \quad i \in (1,2,...,n), \quad \xi_i \geq 0 \tag{31}$$

核函数的选择

对一个具体问题我们需要选择一个核函数以达到获得最好的模型，但如何快速做出选择并没有一个具体的方法，在很多问题上，需要尝试各个核函数，每个核函数中的参数也需要大量尝试，最后经过对比找到一个效果最好的核函数，也就得到最终的分离超曲面和决策函数。

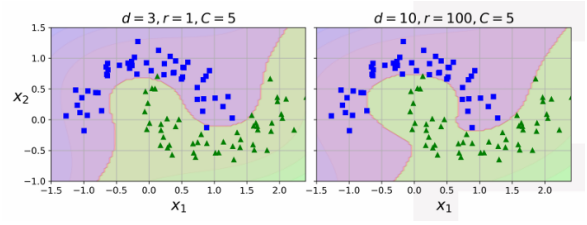
常见核技巧：

$$\text{线性（}Linear\text{）：}\ K(a,b)=a^Tb \tag{32}$$
$$\text{多项式（}Polynomial\text{）：}\ K(a,b)=(\gamma a^Tb+r)^d \tag{33}$$
$$\text{高斯径向基函数（}RBF\text{）：}\ K(a,b)=exp(-\gamma||a-b||^2)\text{（无限维特征空间）} \tag{34}$$
$$Sigmod\text{函数：}\ K(a,b)=tanh(\gamma a^Tb+r) \tag{35}$$

高斯径向基函数（Gaussian Radial Basis Function）是一般通用的近似器。

代码实现（以多项式核为例）

```
from sklearn import svm
clf_poly = svm.SVC(kernel='poly', degree=3, coef0=1, C=5)
clf_poly.fit(x_train, y_train)
#可视化
Visualization(clf_poly, x_train, x_test, y_train, y_test)
```



- d是多项式特征的阶数
- r（即系数0，coef0）是多项式核超参数
- C是软间隔超参数

- 多项式阶数的选择：**欠拟合时请增加阶数，过拟合时请减少阶数。**
- coef0影响高阶项(terms)与低阶项的组合；加coef0的值可能会使得决策边界更加复杂，而减小coef0的值可能会简化决策边界。此参数在多项式核函数和Sigmoid核函数中有效。

支持向量机回归（SVM Regression/SVR）

支持向量机的用途广泛，除了分类，还能做到线性/非线性回归。

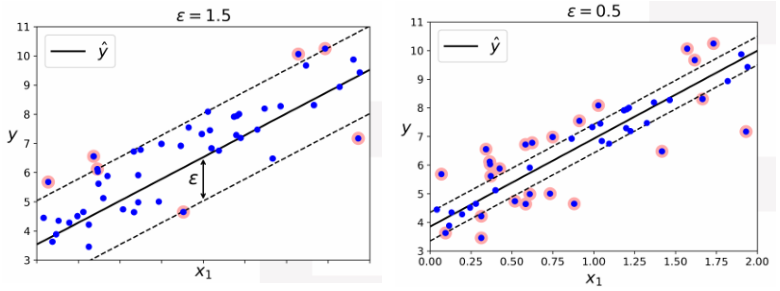
与 SVM分类时希望求出最宽的间隔与尽可能少的间隔内样本 相反，SVM回归（拟合）希望求出尽可能窄的间隔和尽可能少的位于间隔外的样本。

代码实现：

线性回归

超参数ε（epsilon），使用epsilon_insensitive模型

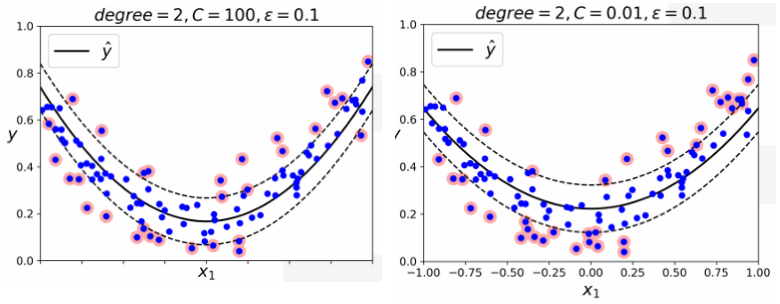
```
from sklearn.svm import LinearSVR
svm_reg = LinearSVR(epsilon=1.5, random_state=42)
svm_reg.fit(X,y)
```



多项式回归

软间隔超参数C，使用多项式核函数

```
from sklearn.svm import LinearSVR
svm_poly_reg = SVR(kernel="ploy", degree=2, C=100, epsilon=0.1, gamma="scal
e")
svm_poly_reg.fit(X,y)
```



参考

- Qoo机器学习笔记

  https://www.zhihu.com/column/c_1222851054658093056

- 【损失函数】Hinge Loss 合页损失

  https://blog.csdn.net/Next_SummerAgain/article/details/135372865

- 拉格朗日对偶问题 - 冷风的文章 - 知乎
  https://zhuanlan.zhihu.com/p/589965451

- 核函数(Kernel function)(举例说明，通俗易懂)

  https://blog.csdn.net/mengjizhiyou/article/details/103437423