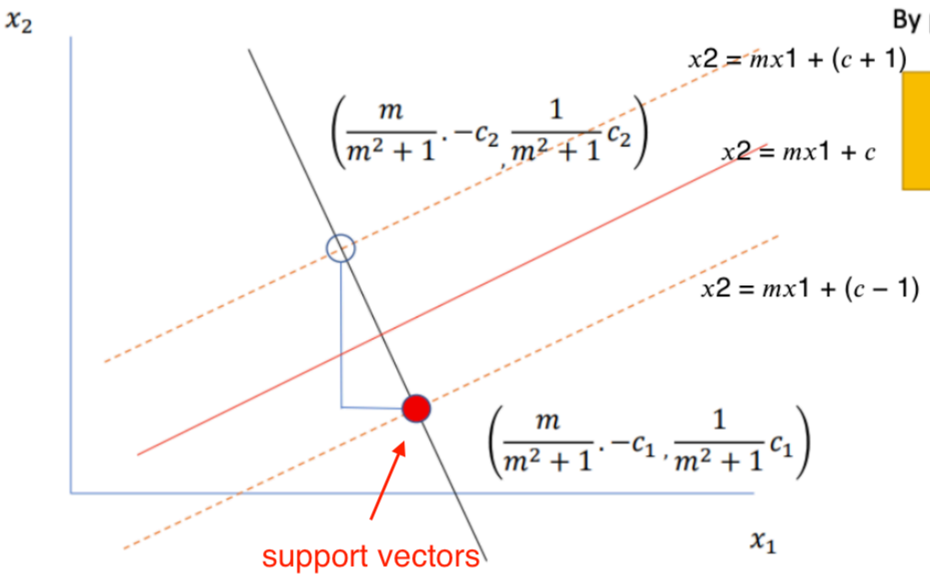
SVM就是试图把线放在最佳位置，好让在线的两边有尽可能大的间隙（几何间隔最大化）。

Finds the boundary with "**maximum margin**"
Uses "**slack variables**" to deal with outliers
Uses "kernels", and the "**kernel trick**", to solve nonlinear problems.

1. Margin (max)



By pythagoras:

$$\boxed{margin = \frac{|c_2 - c_1|}{\sqrt{m^2+1}}}$$

The margin in *wx*-b form

$$\boldsymbol{wx} - b = 0$$

$$w_1x_1 + w_2x_2 - b = 0$$

$$x_2 = \frac{-w_1}{w_2}x_1 + \frac{b}{w_2}$$

$$x_2 = mx_1 + c$$

$$margin = \frac{|c_2 - c_1|}{\sqrt{m^2+1}}$$

Set upper margin with offset b+1, lower margin with offset b-1

Numerator: $c_2 - c_1 = \frac{(b+1)}{w_2} - \frac{(b-1)}{w_2} = \frac{2}{w_2}$

Denominator: $\sqrt{m^2+1} = \sqrt{\left(\frac{-w_1}{w_2}\right)^2 + 1}$

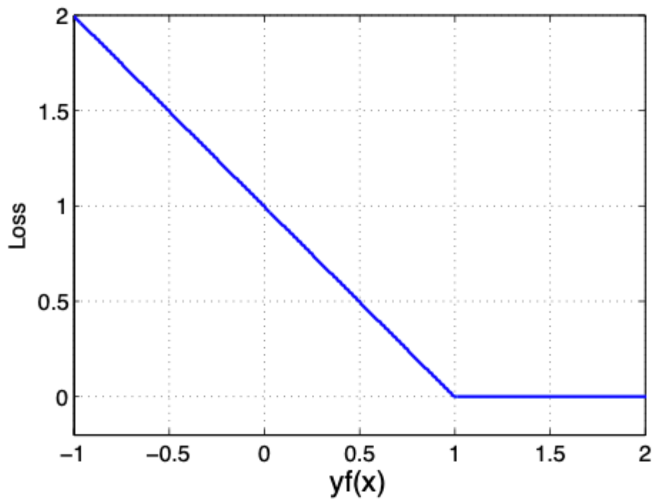
$$= \sqrt{\frac{w_1^2+w_2^2}{w_2^2}} = \frac{|\boldsymbol{w}|}{w_2}$$

$$\boxed{margin = \frac{2}{|\boldsymbol{w}|}}$$

|W| = 平方和根

2. Hinge loss (min)

$$L_{hinge} = \max\left\{0, 1 - y_i f(\mathbf{x}_i)\right\}$$



3. SVM loss function

Margin —> max —> 1/w = x^2
Hinge loss —> min

$$E = \underbrace{\sum_{i=1}^{N} \max\left\{0, 1 - y_i f(\mathbf{x}_i)\right\}}_{\text{hinge loss}} + \underbrace{\frac{1}{2}\sum_{j=1}^{d} w_j^2}_{\text{margin}}$$

---

4. Slack variable —— Soft Margin

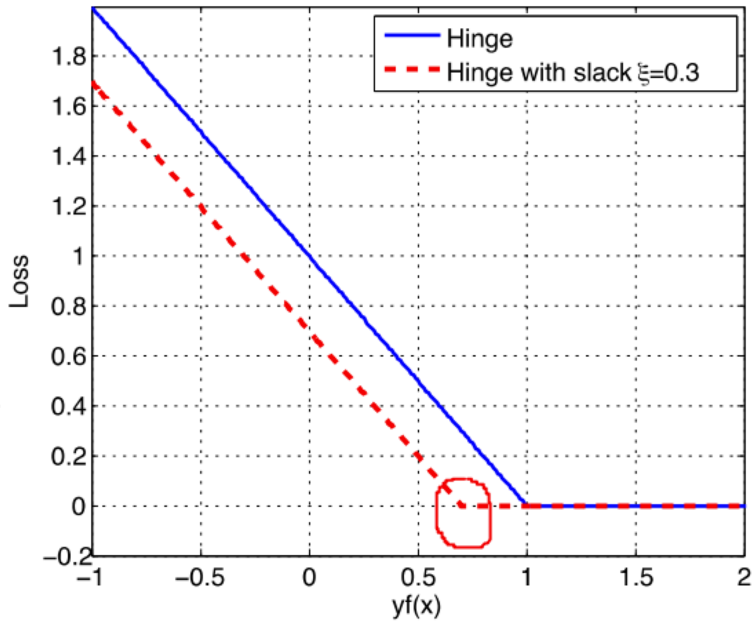
- Key idea: allow some of the data points to be misclassified (outliner)

放宽约束，允许一些并没有正确分类的样本存在

修改 hinge loss

$$L_{hinge} = \max\left\{0, 1 - y_i f(\mathbf{x}_i)\right\}$$

$$L_{hinge} = \max\left\{0, 1 - y_i f(\mathbf{x}_i) - \xi_i\right\},$$



$$E = \sum_{i=1}^{N} \max\left\{0, 1 - y_i f(\mathbf{x}_i) - \underbrace{\xi_i}_{\text{slack}}\right\} + \frac{1}{2}\sum_{j=1}^{d} w_j^2 + \underbrace{C\sum_{i=1}^{N} \xi_i}_{\text{Penalty for using slack – amount of penalty is controlled by a regularisation constant, C}}$$

修改loss function

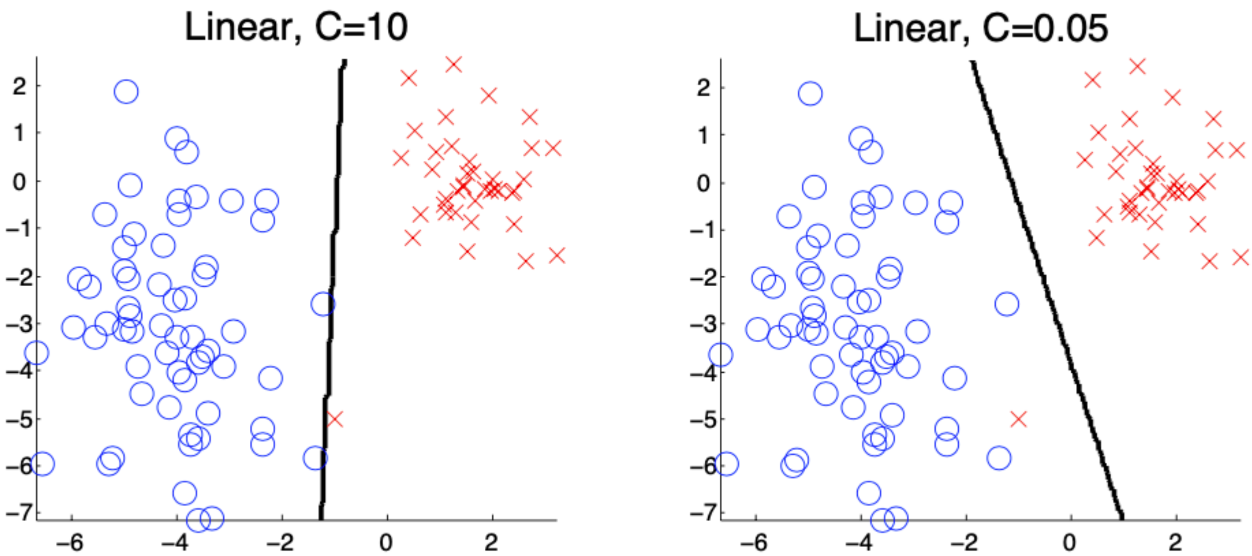
The value of C (slack variable penalty) 它就是权衡误差和间距的参数

roughly translates as how “soft” the margins will be

The default value C = 1

1.A smaller value (right) means some data points are allowed to violate the margins, hence an approximate SVM solution is found, but it has a larger margin
C很小，它会给你一个大间距，但是作为牺牲，我们必须要忽视一些错误分类的样本

2.large value (left) means a very strict penalty, so a very strict SVM solution will be found
C很大，你会尽量正确地分类样本，但是这样做的代价会导致你有很小的间距



因此，在SVM算法的训练上，我们可以通过减小C值来避免overfitting的发生。

1. C大了，松弛变量小，margin小，overfit

2. C小了，margin大

---

5. Non-linear SVM “Kernel Trick”

对于非线性的情况，SVM 的处理方法是选择一个核函数 κ(·,·)，通过将数据映射到高维空间，来解决在原始空间中线性不可分的问题。

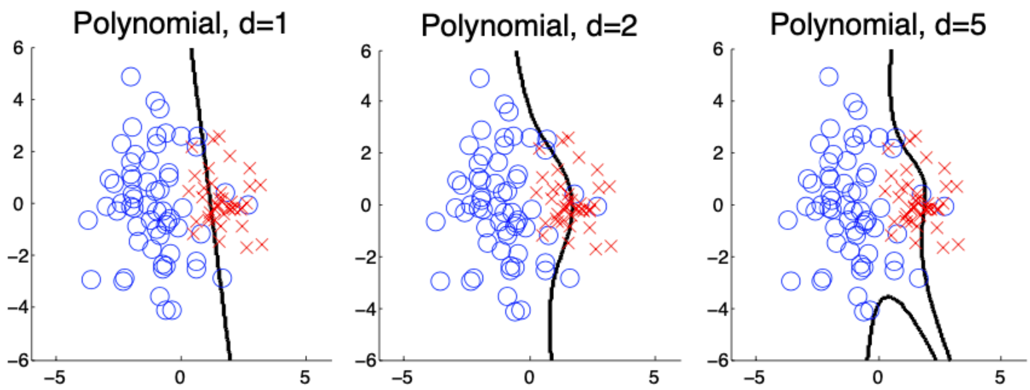
Kernel trick

$$K(\mathbf{x}_i, \mathbf{x}') = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}')$$

$$f(\mathbf{x}') = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}').$$

1.The Polynomial kernel

$$K(\mathbf{x}_i, \mathbf{x}') = (1 + \mathbf{x}_i^T \mathbf{x}')^d$$



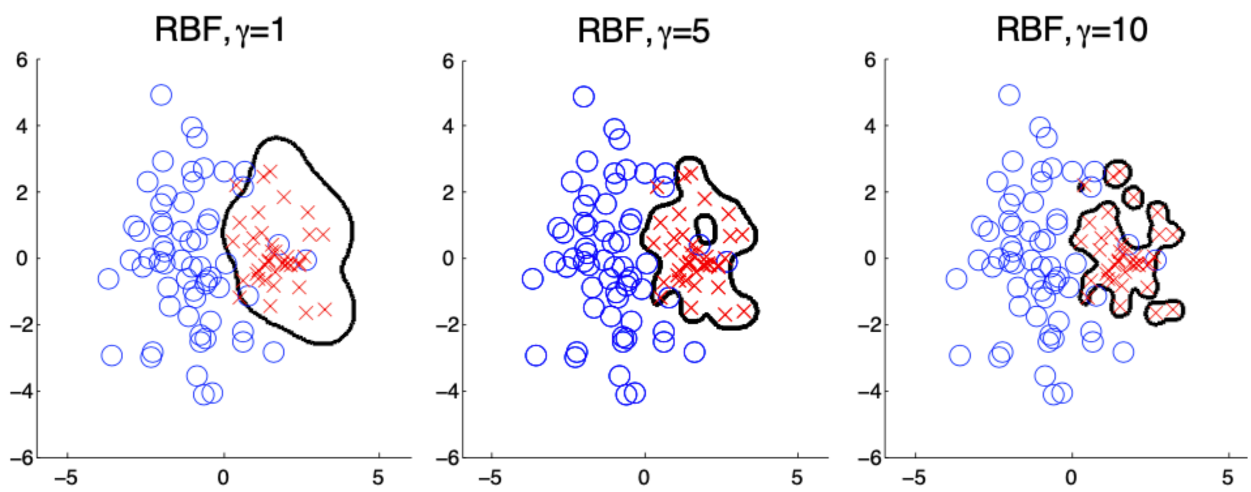
d=1 linear

The higher degree, the more high order terms are introduced to the implicit feature space , hence the final decision boundary becomes more complex.

2.The RBF kernel

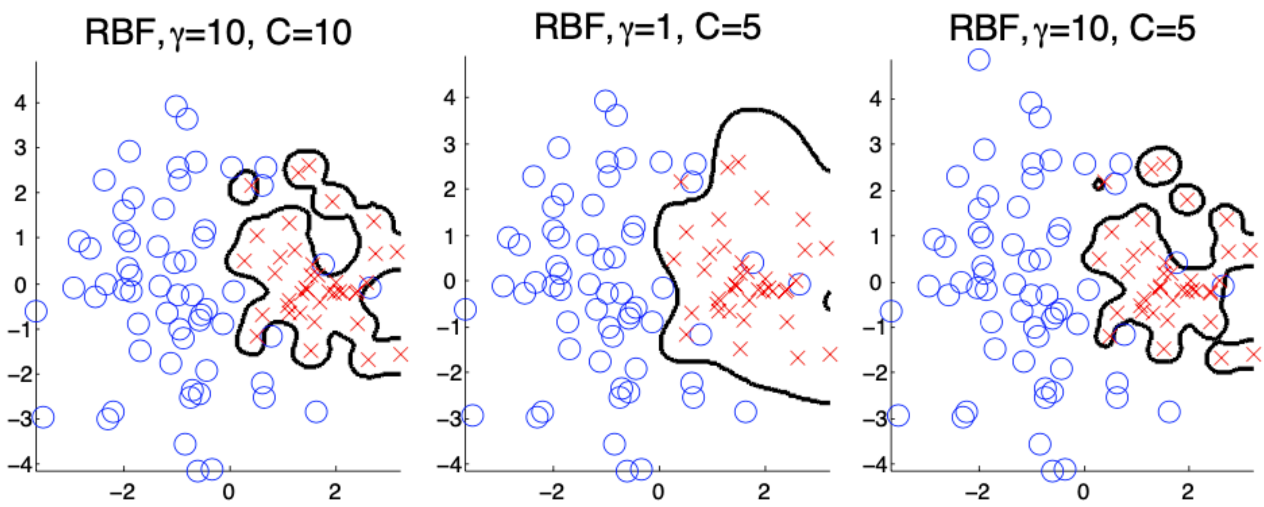
$$K(\mathbf{x_i}, \mathbf{x}') = e^{-\gamma(\mathbf{x}_i - \mathbf{x}')^2} \qquad \gamma = \frac{1}{2\sigma^2}$$ gamma y = standard deviation

The larger γ is, the more overfitting we might expect.
A smaller γ will produce smooth boundaries, possibly underfitting.



the parameters are not independent

so for example, below is what happens when we try to set the C (slack variable penalty) and parameters, at the same time.



输入	含义	解决问题	核函数的表达式	参数 gamma	参数 degree	参数 coef0
"linear"	线性核	线性	$K(x,y)=x^Ty=x\cdot y$	No	No	No
"poly"	多项式核	偏线性	$K(x,y)=(\gamma(x\cdot y)+r)^d$	Yes	Yes	Yes
"sigmoid"	双曲正切核	非线性	$K(x,y)=tanh(\gamma(x\cdot y)+r)$	Yes	No	Yes
"rbf"	高斯径向基	偏非线性	$K(x,y)=e^{-\gamma\|x-y\|^2},\gamma>0$	Yes	No	No

---

K-NN Classifier 非线性模型 一定要标准化normalize

K-nearest neighborhood

**近似误差：训练集的训练误差。
**估计误差：测试集的测试误差。
k=1时，找到的邻居就是自己， training error =0

1.如果选择较小的K值，就相当于用较小的邻域中的训练实例进行预测，学习的**近似误差**会减小，学习的估计误差会增大，整体模型变复杂 **overfit**

Main differences between the perceptron and linear SVM with hard margins:

- SVM is deterministic (Perceptron decision boundary may not always be the same, even for the same data set)
- SVM maximises the margin (Perceptron decision boundary might only 'just' separate the data)

2.如果选择较大K值，就相当于用较大邻域中的训练实例进行预测，其优点是可以减少学习的估计误差，但近似误差会增大，也就是对输入实例预测不准确，K值得增大就意味着整体模型变的简单 **underfit**

留一交叉验证（leave-one-out）：每次从个数为N的样本集中，取出一个样本作为验证集，剩下的N-1个作为训练集，重复进行N次。最后平均N个结果作为泛化误差估计。

K折交叉验证：把数据分成K份，每次拿出一份作为验证集，剩下k-1份作为训练集，重复K次。最后平均K次的结果，作为误差评估的结果。与前两种方法对比，只需要计算k次，大大减小算法复杂度，被广泛应用。