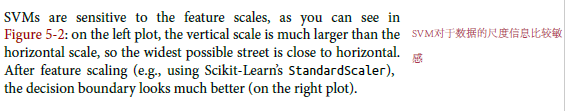
Support Vector Machines

# Linear SVM Classification

基于感知器发展而来的机器学习算法。

SVM: 找出一个超平面能分类数据集，同时使得分割面到距离分割面距离最近的点的距离之和最大。



## Soft Margin Classification

在代价函数中引入松弛变量，将原来的硬边界分类，转化为软边界分类，容许少量的点错分。

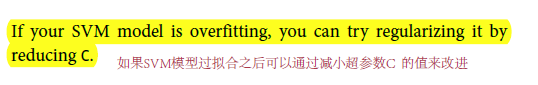
**hard margin classification：**每个样本都被正确的分类对用类中

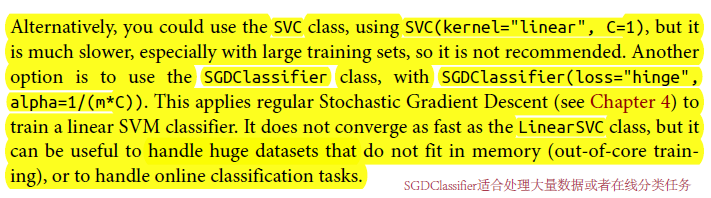
存在两个问题：1、要求数据集线性可分；2、对异常值非常敏感。这造成很难找到一个决策边界，使得SVM算法应用非常局限。

**Soft Margin Classification：**允许少量的点分类错误，在使得街道尽可能宽和尽量少的犯错误中的分类结果，部分实例会在margain之中或者在其错误的一边。

The objective is to find a good balance between keeping the street as large as possible and limiting the margin violations (i.e., instances that end up in the middle of the street or even on the wrong side). This is called soft margin classification.

C值越小，margin越宽，但是错分的点也就越多。In Scikit-Learn’s SVM classes, you can control this balance using the C hyperparameter: a smaller C value leads to a wider street but more margin violations.

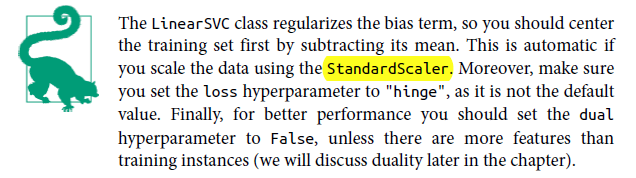




LinearSVC(C=1, loss="hinge"),最快

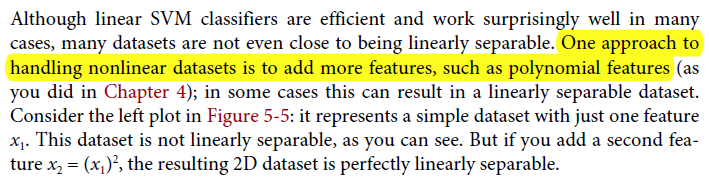
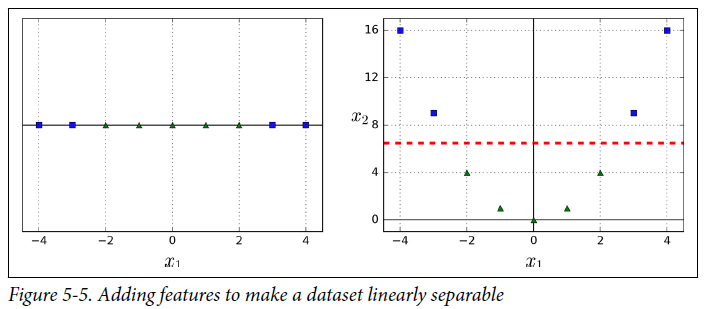
SVC(kearnal = ‘linear’,C = 1)，很慢

SGDClassifier(loss = ‘hinge’,alpha = 1/(m\*C)，适合大数据集



# Nonlinear SVM Classification

非线性解决办法：1采用多项式，也就是空间变换，将输入数据变换到一个线性可分的特征空间。

## 2.1 Polynomial Kernel

SVC(kernel="poly", degree=3, coef0=1, C=5)

通过多项式核空间变换来处理非线性可分数据。

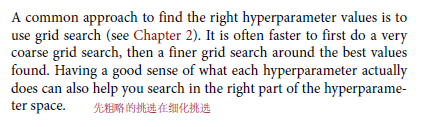
One approach to handling nonlinear datasets is to add more features, such as polynomial features.

通过kernal来进行特征空间变换。

自由度的大小决定了模型的复杂度，就好像多项式回归一样，自由度越大，模型越复杂，从而越容易过拟合。

如果过拟合，就降低多项式核的自由度，相反的，如果欠拟合，就增大多项式核的自由度。

SVM超参数的调整方法：借助grid search 来实现

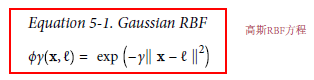


## 2.2 Adding Similarity Features

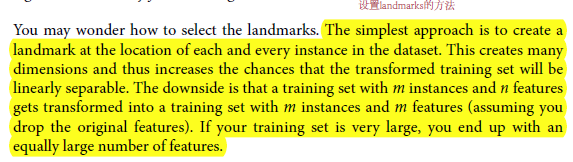
采用相似函数添加特征。

Another technique to tackle nonlinear problems is to add features computed using a similarity function that measures how much each instance resembles a particular landmark.

let’s define the similarity function to be the Gaussian Radial Basis Function (RBF).



如何来设置landmarks

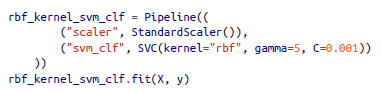


2.2.1 Gaussian RBF Kernel

SVC(kernel="rbf", gamma=5, C=0.001)

添加相似特征适用任何的机器学习算法，但是这回增加计算成本，特别是在很大的数据集上。

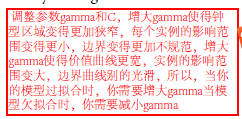
Once again the kernel trick does its SVM magic: it makes it possible to obtain a similar result as if you had added many similarity features, without actually having to add them. Let’s try the Gaussian RBF kernel using the SVC class：



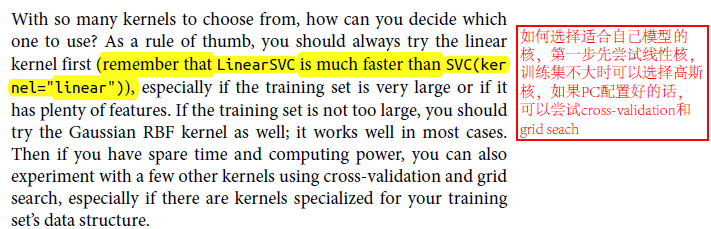
This model is represented on the bottom left of Figure 5-9. The other plots show models trained with different values of hyperparameters gamma (γ) and C. Increasing gamma makes the bell-shape curve narrower (see the left plot of Figure 5-8), and as a result each instance’s range of influence is smaller: the decision boundary ends up being more irregular, wiggling around individual instances. Conversely, a small gamma value makes the bell-shaped curve wider, so instances have a larger range of influence, and the decision boundary ends up smoother. So γ acts like a regularization hyperparameter: if your model is overfitting, you should reduce it, and if it is underfitting, you should increase it (similar to the C hyperparameter).

当γ很大时，钟形结构越狭小，单个实例的影响范围越小，因此决策边界会非常的不规则，也就是非线性更加明显，即模型复杂度更高。

γ对欠拟合与过拟合的影响：如果过拟合，需要减小γ；当欠拟合是需要增大γ；



**如何选择kernel?**



## 2.3 Computational Complexity

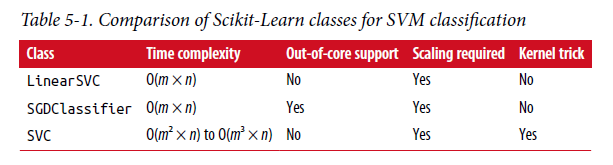
The LinearSVC class is based on the *liblinear* library, which implements an optimized algorithm for linear SVMs.1 It does not support the kernel trick, but it scales almost linearly with the number of training instances and the number of features: its training time complexity is roughly O(m × n).

linearSVC 不支持kernel ,它的计算复杂度大概为O(mXn)

The SVC class is based on the *libsvm* library, which implements an algorithm that supports the kernel trick.2 The training time complexity is usually between O(m2 × n) and O(m3 × n).

SVC支持kernel，它的计算复杂度一般在O(m2 × n) 到O(m3 × n)之间，适合复杂的中小型数据集。

This algorithm is perfect for complex but small or medium training sets. However, it scales well with the number of features, especially with *sparse features* (i.e., when each instance has few nonzero features)



# SVM Regression

As we mentioned earlier, the SVM algorithm is quite versatile: not only does it support linear and nonlinear classification, but it also supports linear and nonlinear regression. The trick is to reverse the objective: instead of trying to fit the largest possible street between two classes while limiting margin violations, SVM Regression tries to fit as many instances as possible *on* the street while limiting margin violations (i.e., instances *off* the street).

替代SVM分类中尽可能少的错分点的情况下使街道更宽，SVM回归尝试在尽可能少的错分点的情况下拟合更多的点。

Adding more training instances within the margin does not affect the model’s predictions; thus, the model is said to be ϵ-insensitive。在街道边缘内增加实例点，不会影响模型的预测。

#使用LinearSVR来做SVM回归

svm\_reg = LinearSVR(epsilon=1.5)

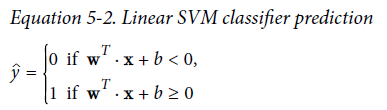
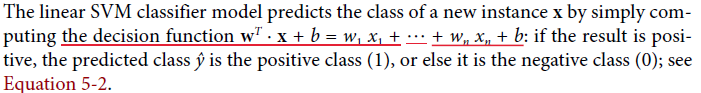
#以多项式为核，使用SVR来做SVM回归

svm\_poly\_reg = SVR(kernel="poly", degree=2, C=100, epsilon=0.1)

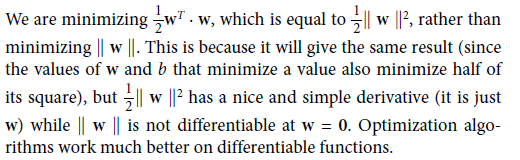
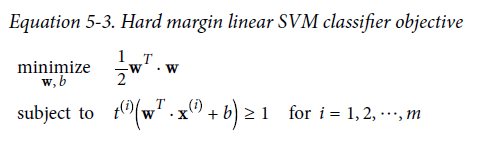
# Under the Hood

SVM怎样预测，SVM的训练算法是怎样工作的？

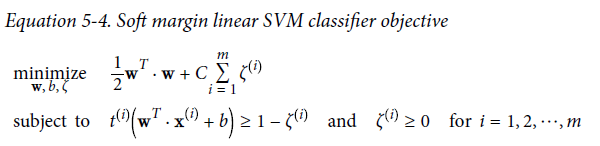
## 4.1 Decision Function and Predictions

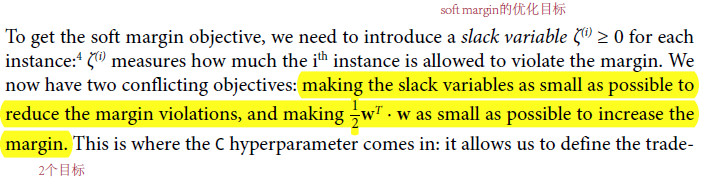


## 4.2 training object



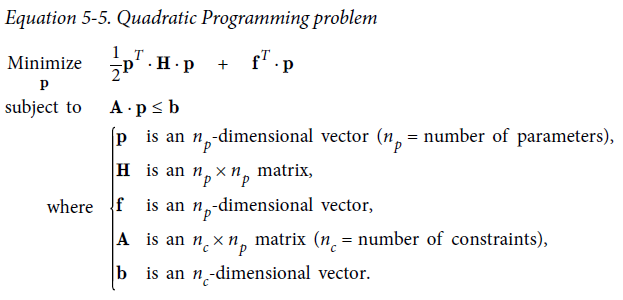
软边缘SVM





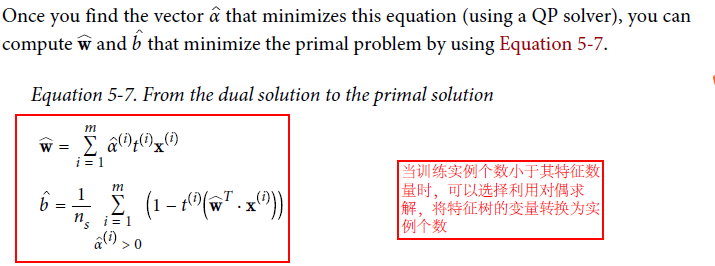
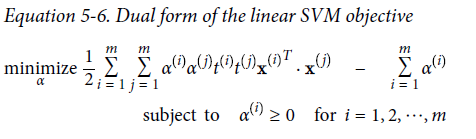
## 4.3 Quadratic Programming

Hard margin 和soft margin问题都是convex quadratic optimization problem with linear constrains. 这样的问题都是二次优化问题。



## 4.4 The Dual Problem

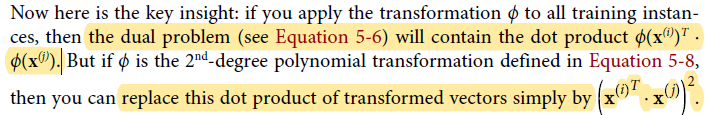
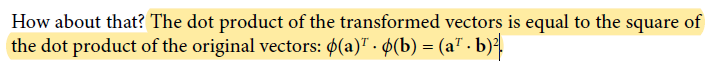
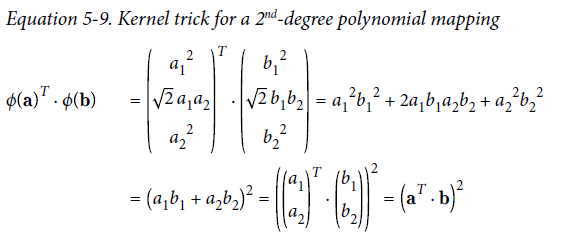
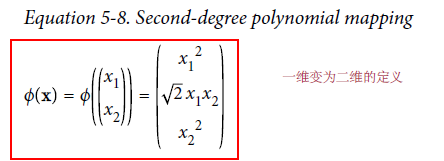
Given a constrained optimization problem, known as the primal problem, it is possible to express a different but closely related problem, called its dual problem



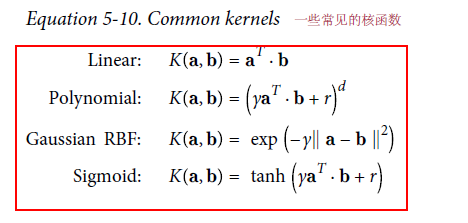
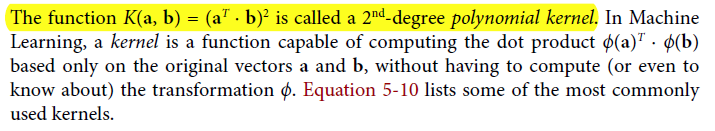
The dual problem is faster to solve than the primal when the number of training instances is smaller than the number of features. More importantly, it makes the kernel trick possible, while the primal does not.

## 4.5 Kernelized SVM

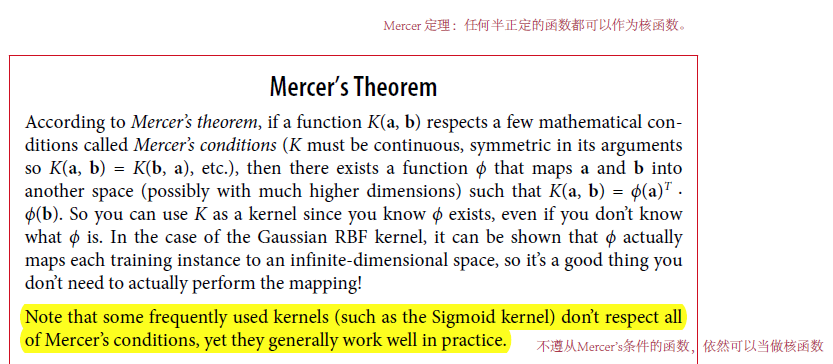
利用数据集的变换函数，将原来的训练集从原来不线性可分的特征空间变换到一个线性可分的特征空间。

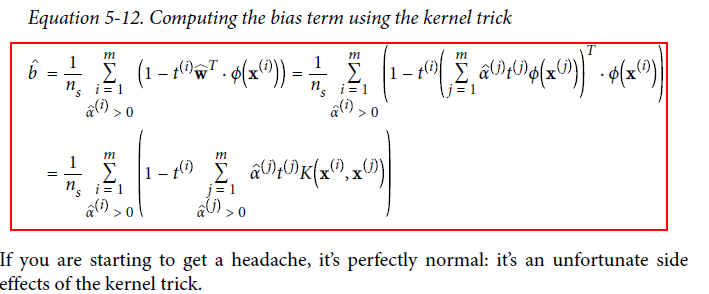
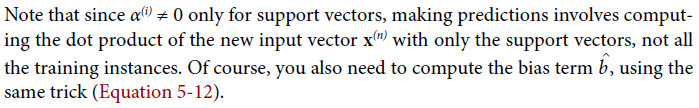
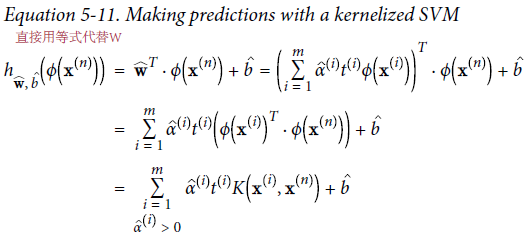


核：



Mercer’s Theorem 表面任何半正定的函数都可以作为核函数，这样其必然存在一个φ函数来经数据集做空间变换。





## 4.6 online SVM

