* **SVM的原理是什么？**

SVM是一种二类分类模型。它的基本模型是在特征空间中寻找间隔最大化的分离超平面的线性分类器。（间隔最大是它有别于感知机）

（1）当训练样本线性可分时，通过硬间隔最大化，学习一个线性分类器，即**线性可分支持向量机**；

（2）当训练数据近似线性可分时，引入松弛变量，通过软间隔最大化，学习一个线性分类器，即**线性支持向量机**；

（3）当训练数据线性不可分时，通过使用核技巧及软间隔最大化，学习**非线性支持向量机**。

注：以上各SVM的数学推导应该熟悉：**硬间隔最大化（几何间隔）---学习的对偶问题---软间隔最大化（引入松弛变量）---非线性支持向量机（核技巧）。**

* **SVM为什么采用间隔最大化？**

当训练数据线性可分时，存在无穷个分离超平面可以将两类数据正确分开。感知机利用误分类最小策略，求得分离超平面，不过此时的解有无穷多个。

线性可分支持向量机**利用间隔最大化求得最优分离超平面**，这时，解是唯一的。另一方面，此时的分隔超平面所产生的分类结果是**最鲁棒**的，对未知实例的**泛化能力最强**。

然后应该借此阐述，几何间隔，函数间隔，及从函数间隔—>求解最小化1/2 ||w||^2 时的w和b。即线性可分支持向量机学习算法—最大间隔法的由来。

* **为什么要将求解SVM的原始问题转换为其对偶问题？**

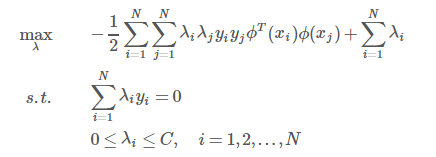
一、是对偶问题往往更易求解（**当我们寻找约束存在时的最优点的时候，约束的存在虽然减小了需要搜寻的范围，但是却使问题变得更加复杂。**为了使问题变得易于处理，我们的方法是**把目标函数和约束全部融入一个新的函数，即拉格朗日函数，再通过这个函数来寻找最优点。**）

二、自然引入核函数，进而**推广到非线性分类问题。**

* **为什么SVM要引入核函数？**

当样本在原始空间线性不可分时，可将样本从原始空间映射到一个更高维的特征空间，使得样本在这个特征空间内线性可分。

引入映射后的对偶问题：



在学习预测中，只定义核函数*K*(*x*,*y*)，而不是显式的定义映射函数*ϕ*。因为**特征空间维数可能很高，甚至可能是无穷维，因此直接计算*ϕ*(*x*)·*ϕ*(*y*)是比较困难的。**相反，直接计算*K*(*x*,*y*)比较容易（即直接在原来的低维空间中进行计算，而不需要显式地写出映射后的结果）。

核函数的定义：*K*(*x*,*y*)=<*ϕ*(*x*),*ϕ*(*y*)>，即在特征空间的内积等于它们在原始样本空间中通过核函数K计算的结果。

除了 SVM 之外，任何**将计算表示为数据点的内积的方法，都可以使用核方法进行非线性扩展。**

* **RBF核函数的具体公式？**

**https://img-blog.csdn.net/20160828185625701**

**Gauss径向基函数则是局部性强的核函数，其外推能力随着参数σ的增大而减弱。**

**这个核会将原始空间映射为无穷维空间。**不过，如果 *σ* 选得很大的话，**高次特征上的权重实际上衰减得非常快，所以实际上（数值上近似一下）相当于一个低维的子空间；** 反过来，如果 *σ* 选得很小，则可以将任意的数据映射为线性可分——当然，**这并不一定是好事，因为随之而来的可能是非常严重的过拟合问题。**不过，总的来说，通过调控参数*σ* ，**高斯核实际上具有相当高的灵活性，也是使用最广泛的核函数之一。**

* **为什么SVM对特征缺失数据敏感？**

这里说的缺失数据是指**缺失某些****特征数据**，向量数据不完整。SVM没有**处理缺失值的策略**（决策树有）。而**SVM希望样本在特征空间中线性可分，所以特征空间的好坏对SVM的性能很重要**。缺失特征数据将影响训练结果的好坏。

其实就是，每个特征都会被用到，SVM希望样本在样本空间内，是可分的，因此特征的缺失对影响到样本空间的划分。

* **SVM对噪声敏感原因？**

SVM的决策面可以认为是由**决策面附近的支持向量决定的，如果分类面附近存在噪声，势必会影响决策面特性。**SVM如果将噪声视为支持向量，则SVM会拟合噪声的性质，造成模型的过拟合。例如，因为一两个噪声点的存在，导致原本线性可分的问题变成线性不可分的，则模型会变得复杂，我们可能需要使用核函数，来将样本映射到高维空间，使其线性可分。

* **SVM是用的是哪个库？Sklearn/libsvm中的SVM都有什么参数可以调节？**

用的是sklearn实现的。采用**sklearn.svm.SVC**设置的参数。本身这个函数也是基于libsvm实现的（PS: libsvm中的二次规划问题的**解决算法是SMO**）。

**SVC函数的训练时间是随训练样本平方级增长，所以不适合超过10000的样本。**

对于多分类问题，SVC采用的是**one-vs-one投票机制**，需要两两类别建立分类器，训练时间可能比较长。

**sklearn.svm.SVC(*C=1.0*, *kernel='rbf'*, *degree=3*, *gamma='auto'*, *coef0=0.0*, *shrinking=True*, *probability=False*,*tol=0.001*, *cache\_size=200*, *class\_weight=None*, *verbose=False*, *max\_iter=-1*, *decision\_function\_shape=None*,*random\_state=None*)**

**参数：**

l  C：C-SVC的惩罚参数C?默认值是1.0

**C越大，相当于惩罚松弛变量，希望松弛变量接近0，即对误分类的惩罚增大，趋向于对训练集全分对的情况，这样对训练集测试时准确率很高，但泛化能力弱。C值小，对误分类的惩罚减小，允许容错，将他们当成噪声点，泛化能力较强。**

l  **kernel** ：核函数，默认是rbf，可以是‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’

  　　0 – 线性：u'v ==线性可分的情况

 　　 1 – 多项式：(gamma\*u'\*v + coef0)^degree

  　　2 – RBF函数：exp(-gamma|u-v|^2)

  　　3 – sigmoid：tanh (gamma\*u'\*v + coef0)

l  **degree** ：多项式poly函数的维度，默认是3，选择其他核函数时会被忽略。

l  **gamma** ： ‘rbf’, ‘poly’ 和‘sigmoid’的核函数参数。默认是’auto’，则会选择1/n\_features

l  **coef0** ：核函数的常数项。对于‘poly’和 ‘sigmoid’有用。

l  **probability ：**是否采用概率估计？.默认为False

l  **shrinking** ：是否采用shrinking heuristic方法，默认为true

l  **tol ：**停止训练的误差值大小，默认为1e-3

l  **cache\_size** ：核函数cache缓存大小，默认为200

l  **class\_weight** ：类别的权重，字典形式传递。设置第几类的参数C为weight\*C(C-SVC中的C)

l  **verbose** ：允许冗余输出

l  **max\_iter** ：最大迭代次数。-1为无限制。

l  **decision\_function\_shape** ：‘ovo’, ‘ovr’ or None, default=None

l  **random\_state** ：数据洗牌时的种子值，int值

主要调节的参数有：C、kernel、degree、gamma、coef0。

**C越大，gamma越大，越容易过拟合。**

0 -- **linear**: u'\*v  
1 -- polynomial: (gamma\*u'\*v + coef0)^degree  
2 -- **radial basis function**: exp(-gamma\*|u-v|^2)  
3 -- sigmoid: tanh(gamma\*u'\*v + coef0)

使用**Grid Search**，两两组合调参。

* **如何选择核函数？**

（1）**如果特征维数很高，往往线性可分（SVM解决非线性分类问题的思路就是将样本映射到更高维的特征空间中），可以采用LR或者线性核的SVM；**

（2）如果样本数量很多，由于求解最优化问题的时候，目标函数涉及两两样本计算内积，使用高斯核明显计算量会大于线性核，所以手动添加一些特征，使得线性可分，然后可以用LR或者线性核的SVM；

（3）如果不满足上述两点，即特征维数少，样本数量正常，可以使用高斯核的SVM。

* **SVM如何处理多分类问题？**

对训练器进行组合。其中比较典型的有**一对一**，和**一对多**。

一对多，就是对每个类都训练出一个分类器，由svm是二分类，所以将此而分类器的两类设定为目标类为一类，其余类为另外一类。这样针对k个类可以训练出k个分类器，当有一个新的样本来的时候，用这k个分类器来测试，那个分类器的概率高，那么这个样本就属于哪一类。这种方法效果不太好，bias比较高。

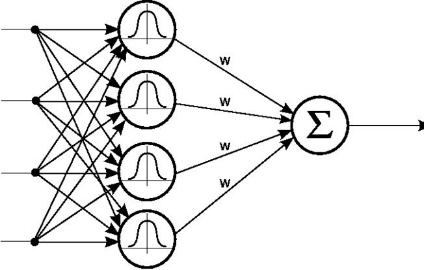
svm一对一法（one-vs-one），针对任意两个类训练出一个分类器，如果有k类，一共训练出C(2,k) 个分类器，这样当有一个新的样本要来的时候，用这C(2,k) 个分类器来测试，每当被判定属于某一类的时候，该类就加一，最后票数最多的类别被认定为该样本的类。

* **LDA与线性核支持向量机的在何种条件下是等价的？**

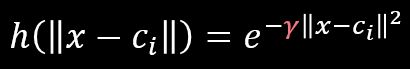
LDA的投影面与线性核SVM的决策超平面是垂直的。

* **高斯核SVM与RBF神经网络之间的联系？**

RBF Network 通常只有三层。输入层、中间层计算输入 x 矢量与样本矢量 c 欧式距离的 Radial Basis Function (RBF) 的值，输出层算它们的线性组合。



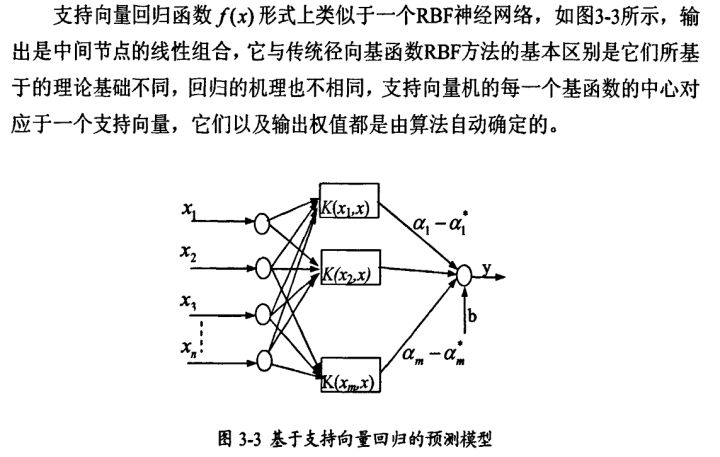
和普通的三层神经网络，RBF 神经网络的区别在中间层。中间层**采用 RBF Kernel 对输入作非线性变换，以便输出层训练线性分类器**。



RBF 拥有较小的支集。针对选定的样本点，它只对样本附近的输入有反应。

**RBF Network 确实与 SVM with RBF kernel 无异。区别在于训练方式。**其实在深度学习出现之前，RBF神经网络就已经提出了 2-stage training。第一阶段为非监督学习，从数据中选取记忆样本（图四中的紫色中心）。例如聚类算法可在该阶段使用。第二阶段为监督学习，训练记忆样本与样本输出的联系。该阶段根据需要可使用 AD/BP。小结：1. RBF神经网络的训练过程可以使用BP，因此应纳入BP神经网络的范畴。2. RBF神经网络的训练分两阶段，这一点类似深度学习，且使RBF神经网络不同于SVM with RBF kernel。

二者的测试过程，基本认为是一致的。都可以认为只有部分点起作用。



* **设计一个能显著减少SVM中支持向量数目而不显著降低泛化性能的方法。**

对样本进行噪声剔除，特别是边缘点。

* **SVM与LR的联系和区别？**

线性可分的支持向量机和线性支持向量机、LR都是线性分类器；

但是存在很大的区别:

1、所采取的策略不一样，前者是几何，后者概率，前者距离后者最大似然函数；

2、SVM部分点决定，LR所有点考虑，随着距离指数衰减；

3、SVM二分类，LR可以多分类。