

数据分析【机器学习】总结之-----SVM(支持向量机)十大常见面试题整理_天阑之蓝的博客-程序员资料 - 程序员资料

阅读之前看这里：博主是正在学习数据分析的一员，博客记录的是在学习过程中一些总结，也希望和大家一起进步，在记录之时，未免存在很多疏漏和不全，如有问题，还请私聊博主指正。

博客地址：[天阑之蓝的博客](#)，学习过程中不免有困难和迷茫，希望大家都能在这学习的过程中肯定自己，超越自己，最终创造自己。

目录

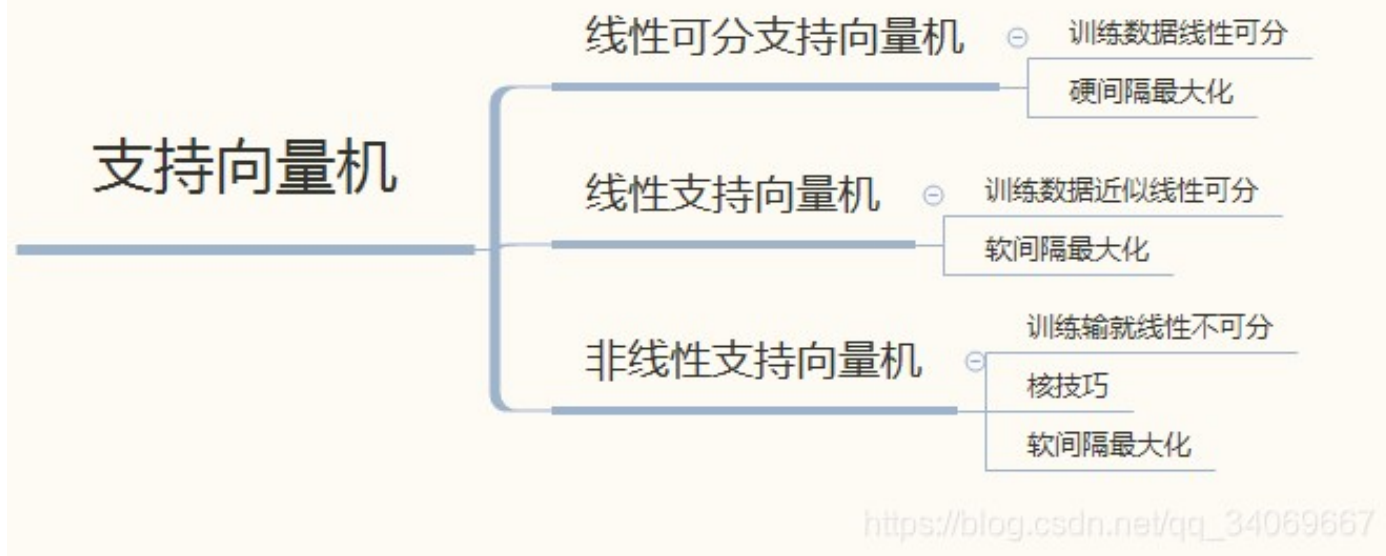
- - [\(1\) SVM的原理是什么？](#)
 - [\(2\) 支持向量机的分类及区别](#)
 - [\(3\) SVM为什么采用间隔最大化？](#)
 - [\(4\) 为什么SVM要引入核函数？\(将线性不可分变成线性可分\)](#)
 - [\(5\) 常用的核函数](#)
 - [\(6\) 为什么要将求解SVM的原始问题转换为其对偶问题？](#)
 - [\(7\) 为什么SVM对缺失数据敏感？](#)
 - [\(8\) SVM如何处理多分类问题？](#)
 - [\(9\) SVM怎么防止过拟合？](#)
 - [\(10\) SMO算法的作用](#)
 - [\(11\) SVM的优缺点](#)

(1) SVM的原理是什么？

支持向量机是一种二分类模型，它的基本模型定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；支持向量机还包括核技巧，这使他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，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最大化问题。支持向量机的学习算法是求解凸二次规划的最优化算法。

简单来说SVM就是，在特征空间上找到最优超平面使得数据的正负样本间隔最大。

(2) 支持向量机的分类及区别



(3) SVM为什么采用间隔最大化?

支持向量机学习的基本思想是求解能够正确划分训练集并且几何间隔最大的最大超平面。对线性可分数据集而言，线性可分分离超平面有无穷多个，但是几何间隔最大的分离超平面是唯一的。

间隔最大化的直观解释是：对训练集找到几何间隔最大的超平面意味着以充分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，而且对最难分的实例点(离超平面最近的点**)也有最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(具有良好的鲁棒性)。

(4) 为什么SVM要引入核函数？(将线性不可分变成线性可分)

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。

(5) 常用的核函数

- 多项式核函数：

$$K(x, z) = (x \cdot z + 1)^p \quad K(x, z) = (x \cdot z + 1)^p \quad K(x, z) = (x \cdot z + 1)^p$$

对应的支持向量机是一个 p 次多项式分类器。

多项式核函数可以实现将低维的输入空间映射到高维的特征空间，但是多项式核函数的参数多，当多项式的阶数比较高的时候，核矩阵的元素值将趋于无穷大或者无穷小，计算复杂度会大到无法计算。

- 高斯核函数：

$$K(x, z) = \exp(-\frac{||x - z||^2}{2\sigma^2}) \quad K(x, z) = \exp(-\frac{||x - z||^2}{2\sigma^2}) \quad K(x, z) = \exp(-\frac{||x - z||^2}{2\sigma^2})$$

$$K(x, z) = \exp(-2\sigma^2 ||x - z||^2)$$

对应的支持向量机是高斯径向基函数分类器。

高斯径向基函数是一种局部性强的核函数，其可以将一个样本映射到一个更高维的空间内，该核函数是应用最广的一个，无论大样本还是小样本都有比较好的性能，而且其相

对于多项式核函数参数要少，因此大多数情况下在不知道用什么核函数的时候，优先使用高斯核函数。

- 线性核函数

$$K(x, z) = x \cdot z \quad K(x, z) = x \cdot z \quad K(x, z) = x \cdot z$$

线性核，主要用于线性可分的情况。我们可以看到特征空间到输入空间的维度是一样的，其参数少|速度快。对于线性可分数据，其分类效果很理想，因此我们通常首先尝试用线性核函数来做分类，看看效果如何，如果不行再换别的

- 神经元的非线性作用核函数 Sigmoid:

$$K(x, z) = \tanh(\eta(x \cdot z) + \theta) \quad K(x, z) = \tanh(\eta(x \cdot z) + \theta) \quad K(x, z) = \tanh(\eta(x \cdot z) + \theta)$$

采用sigmoid核函数，支持向量机实现的就是一种多层神经网络。

因此，在选用核函数的时候，如果我们对我们的数据有一定的先验知识，就利用先验来选择符合数据分布的核函数；如果不知道的话，通常使用交叉验证的方法，来试用不同的核函数，误差最小的即为效果最好的核函数，或者也可以将多个核函数结合起来，形成混合核函数。在吴恩达的课上，也曾经给出过一系列的选择核函数的方法：

如果特征的数量大到和样本数量差不多，则选用**LR**或者线性核的**SVM**；

如果特征的数量小，样本的数量正常，则选用**SVM**+高斯核函数；

如果特征的数量小，而样本的数量很大，则需要手工添加一些特征从而变成第一种情况。

SVM 核函数之间的区别：

一般选择线性核和高斯核，线性核主要用于线性可分的情形，参数少，速度快。

RBF核：主要用于线性不可分的情况，参数多，分类结果非常依赖于参数。有很多人是通过训练数据的交叉验证来寻找合适的参数，不过比较耗时，

（6）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。）

之所以说换为对偶问题更容易求解，其原因在于降低了算法的计算复杂度。在原问题下，算法的复杂度与样本维度相关，即等于权重 w 的维度，而在对偶问题下，算法复杂度与样本数量有关，即为拉格朗日算子的个数。

因此，如果你是做线性分类，且样本维度低于样本数量的话，在原问题下求解就好了，

Liblinear之类的线性**SVM**默认都是这样做的；但如果你是做非线性分类，那就会涉及到升维（比如使用高斯核做核函数，其实是将样本升到无穷维），升维后的样本维度往往会远大于样本数量，此时显然在对偶问题下求解会更好。

另一方面，我们有分析过，只有在支持向量上的样本对应的拉格朗日算子 λ 才大于0，其余的 λ 都是=0，而转为对偶问题的计算对象仅有 λ ，所以大大降低了计算复杂度。

二、自然引入核函数，进而推广到非线性分类问题。

SVM从原始问题变为对偶问题来求解的原因(简化)

1. 对偶问题将原始问题中的约束转为了对偶问题中的等式约束
2. 方便核函数的引入
3. 改变了问题的复杂度。由求特征向量 w 转化为求比例系数 a ，在原始问题下，求解的复杂度与样本的维度有关，即 w 的维度。在对偶问题下，只与样本数量有关。

（7）为什么**SVM**对缺失数据敏感？

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。

(8) SVM如何处理多分类问题？

一般有两种做法：一种是直接法，直接在目标函数上修改，将多个分类面的参数求解合并到一个最优化问题里面。看似简单但是计算量却非常的大。

另外一种做法是间接法：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。

(9) SVM怎么防止过拟合？

如果支持向量中碰巧存在异常点，那么我们傻傻地让SVM去拟合这样的数据，最后的超平面就不是最优。

解决过拟合的办法是为SVM引入了松弛变量 ξ （slack variable）。

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad y_i(w^T x_i + b) \geq 1 - \xi_i$$

因此SVM公示中的目标函数也需要相应修改，我们加上松弛变量的平方和，并求最小值。这样就达到一个平衡：既希望松弛变量存在以解决异常点问题，又不希望松弛变量太大导致分类解决太差。

(10) SMO算法的作用

支持向量机的学习问题可以转化为求解凸二次规划的问题，这样的凸二次规划问题具有全局最优解，并且有许多优化算法可以用于求解这一问题。但是当训练样本容量很大时，这些算法往往就变得低效，以至于无法使用。

序列最小优化算法(SMO)就是解凸二次规划的对偶问题：

SMO算法是一种启发式算法，它是支持向量机的一种快速算法，其特点是不断地将二次规划问题分解为只有两个变量地二次规划问题，并对子问题进行解析求解，直到所有变量满足KKT条件为止。这样通过启发式地方法得到原二次规划问题地最优解。因为子问题有解析解，所以每次计算子问题都很快，虽然子问题次数很多，但在总体上还是高效地。

(11) SVM的优缺点

• SVM的优点：

- 能够处理大型特征空间
- 能够处理非线性特征之间的相互作用
- 无需依赖整个数据，只需依赖支持向量
- 占用内存小，因为不需要保留所有样本，只需要保留支持向量即可，能够获得一个较好的效果。
- 泛化性能良好，学习效果具有较好的推广性

- SVM的缺点:

- 当观测样本很多时，效率并不是很高
- 有时候很难找到一个合适的核函数

为此，我试着编写一个简单的工作流，决定应该何时选择这三种算法，流程如下：

- 首当其冲应该选择的的就是逻辑回归，如果它的效果不怎么样，那么可以将它的结果作为基准来参考；
- 然后试试决策树（随机森林）是否可以大幅度提升模型性能。即使你并没有把它当做最终模型，你也可以使用随机森林来移除噪声变量；
- 如果特征的数量和观测样本特别多，那么当资源和时间充足时，使用SVM不失为一种选择。

参考：[数据挖掘（机器学习）面试-SVM面试常考问题\(https://blog.csdn.net/szlcw1/article/details/52259668\)](https://blog.csdn.net/szlcw1/article/details/52259668)