

机器学习算法系列（12）：SVM（6） — SVM与LR的异同

本文讲述LR与SVM的异同点

一、LR与SVM的相同点

1. LR和SVM都是分类算法，都是监督学习算法。
2. 如果不考虑核函数，LR和SVM都是线性分类算法，也就是说他们的分类决策面都是线性的。LR也是可以用核函数的，至于为什么通常在SVM中运用核函数而不在LR中运用，后面讲到他们之间区别的时候会重点分析。总之，原始的LR和SVM都是线性分类器，这也是为什么通常没人问你决策树和LR什么区别，决策树和SVM什么区别，你说一个非线性分类器和一个线性分类器有什么区别？
3. LR和SVM都是判别模型。判别模型会生成一个表示 $P(Y|X)$ 的判别函数（或预测模型），而生成模型先计算联合概率 $p(Y,X)$ 然后通过贝叶斯公式转化为条件概率。简单来说，在计算判别模型时，不会计算联合概率，而在计算生成模型时，必须先计算联合概率。或者这样理解：生成算法尝试去找到底这个数据是怎么生成的（产生的），然后再对一个信号进行分类。基于你的生成假设，那么那个类别最有可能产生这个信号，这个信号就属于那个类别。判别模型不关心数据是怎么生成的，它只关心信号之间的差别，然后用差别来简单对给定的一个信号进行分类。常见的判别模型有：KNN、SVM、LR，常见的生成模型有：朴素贝叶斯，隐马尔可夫模型。当然，这也是为什么很少有人问你朴素贝叶斯和LR以及朴素贝叶斯和SVM有什么区别。
4. LR和SVM在学术界和工业界都广为人知并且应用广泛。

二、LR与SVM的不同点

1. 损失函数

我们先来看一下带松弛变量的 SVM 和正则化的逻辑回归它们的损失函数：

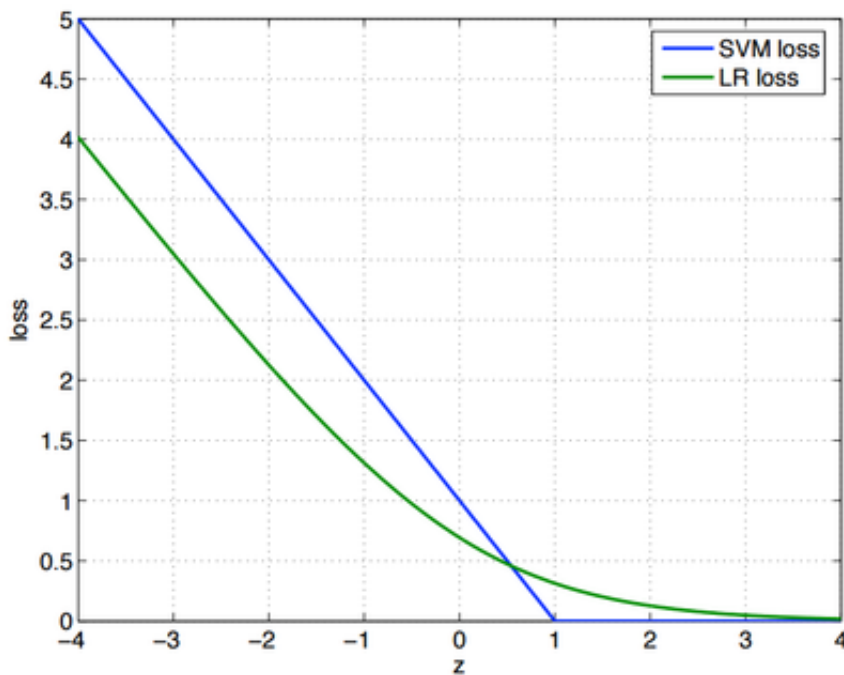
$$\text{SVM: } \frac{1}{n} \sum_{i=1}^n (1 - y_i [w_0 + \mathbf{x}_i^T \mathbf{w}_1])^+ + \lambda \|\mathbf{w}_1\|/2 \quad (1)$$

$$\text{Logistic: } \frac{1}{n} \sum_{i=1}^n \overbrace{-\log g(y_i [w_0 + \mathbf{x}_i^T \mathbf{w}_1])}^{-\log P(y_i | \mathbf{x}, \mathbf{W})} + \lambda \|\mathbf{w}_1\|/2 \quad (2)$$

其中 $g(z) = (1 + \exp(-z))^{-1}$

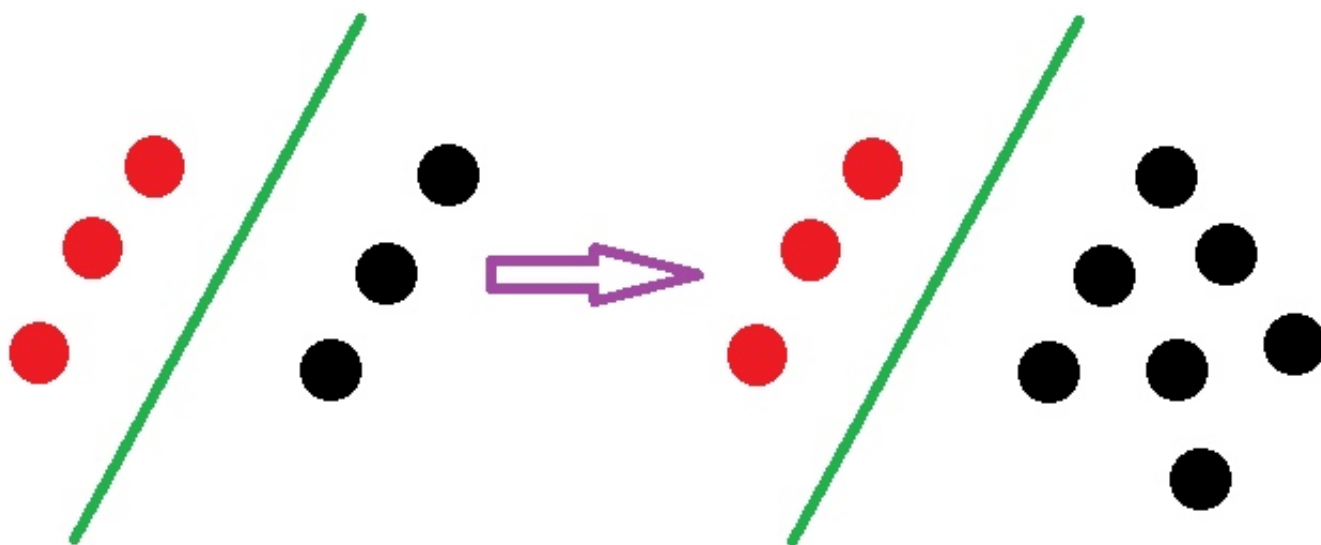
$$\text{Both: } \frac{1}{n} \sum_{i=1}^n \overbrace{\text{Loss}(y_i [w_0 + \mathbf{x}_i^T \mathbf{w}_1])}^z + \lambda \|\mathbf{w}_1\|/2$$

可以将两者统一起来:

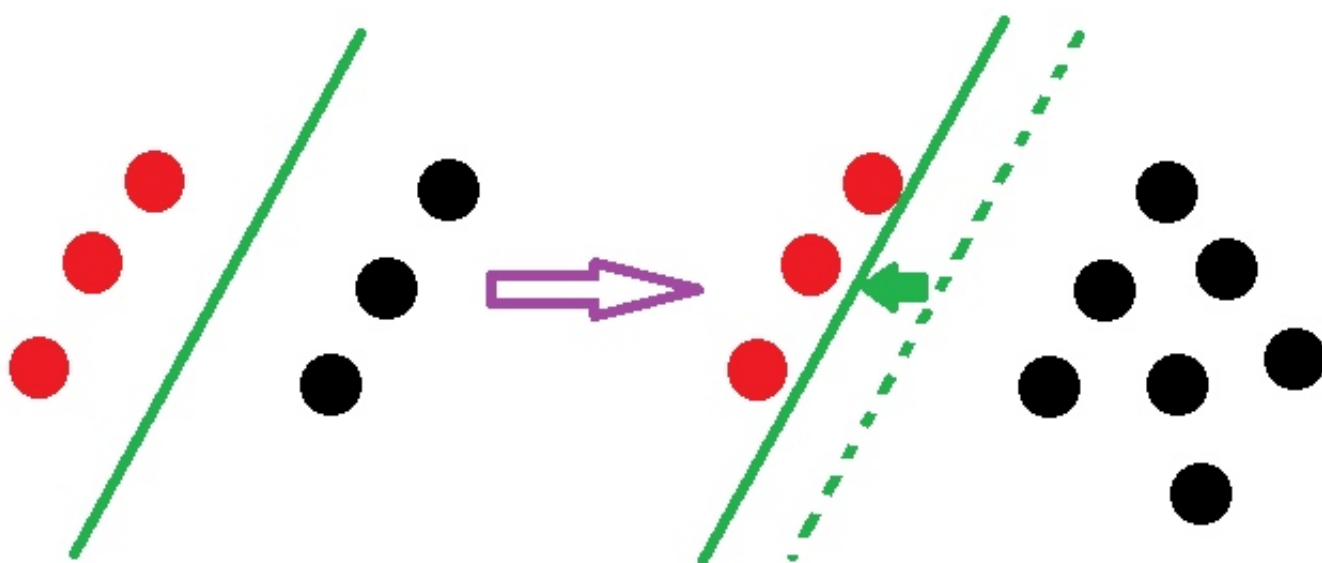


这两个损失函数的目的都是增加对分类影响较大的数据点的权重，减少与分类关系较小的数据点的权重。SVM的处理方法是只考虑support vectors，也就是和分类最相关的少数点，去学习分类器。而逻辑回归通过非线性映射，大大减小了离分类平面较远的点的权重，相对提升了与分类最相关的数据点的权重,两者的根本目的都是一样的。即支持向量机只考虑局部的边界线附近的点，而逻辑回归考虑全局（远离的对边界线的确定也起作用）。

影响SVM决策面的样本点只有少数的支持向量，当在支持向量外添加或减少任何样本点对分类决策面没有任何影响；而在LR中，每个样本点都会影响决策面的结果。用下图进行说明：



支持向量机改变非支持向量样本并不会引起决策面的变化



逻辑回归中改变任何样本都会引起决策面的变化

因此线性SVM不直接依赖于数据分布，分类平面不受一类点影响；LR则受所有数据点的影响，如果数据不同类别strongly unbalance，一般需要先对数据做平衡处理。

2.核技巧

在解决非线性问题时，支持向量机采用核函数的机制，而LR通常不采用核函数的方法。

这个问题理解起来非常简单。分类模型的结果就是计算决策面，模型训练的过程就是决策面的计算过程。通过上面的第二点不同点可以了解，在计算决策面时，SVM转化为对偶问题后，只有少数几个代表支持向量的样本参与了计算，也就是只有少数几个样本需要参与核计算（即kernel machine解的系数是稀疏的），这个在进行复杂核函数计算时优势很明显，能够大大简化模型和计算量。。然而，LR算法里，每个样本点都必须参与决策面的计算过程，也就是说，假设我们在LR里也运用核函数的原理，那么每个样本点都必须参与核计算，这带来的计算复杂度是相当高的。所以，在具体应用时，LR很少运用核函数机制。

3.正则项

根据需要，两个方法都可以增加不同的正则化项，如 l_1, l_2 等等。所以在很多实验中，两种算法的结果是很接近的。但是逻辑回归相对来说模型更简单，好理解，实现起来，特别是大规模线性分类时比较方便。而SVM的理解和优化相对来说复杂一些。但是SVM的理论基础更加牢固，有一套结构化风险最小化的理论基础，虽然一般使用的人不太会去关注。

4.异常值

两者对异常的敏感度也不一样。同样的线性分类情况下，如果异常点较多的话，无法剔除，首先LR，LR中每个样本都是有贡献的，最大似然后会自动压制异常的贡献，SVM+软间隔对异常还是比较敏感，因为其训练只需要支持向量，有效样本本来就不高，一旦被干扰，预测结果难以预料。

5.normalization

两个模型对数据和参数的敏感程度不同，Linear SVM比较依赖penalty的系数和数据表达空间的测度，而（带正则项的）LR比较依赖对参数做L1 regularization的系数。但是由于他们或多或少都是线性分类器，所以实际上对低维度数据overfitting的能力都比较有限，相比之下对高维度数据，LR的表现会更加稳定，为什么呢？

因为Linear SVM在计算margin有多“宽”的时候是依赖数据表达上的距离测度的，换句话说如果这个测度不好（badly scaled，这种情况在高维数据尤为显著），所求得的所谓Large margin就没有意义了，这个问题即使换用kernel trick（比如用Gaussian kernel）也无法完全避免。所以使用Linear SVM之前一般都需要先对数据做normalization，而求解LR（without regularization）时则不需要或者结果不敏感。

Linear SVM 和 LR 有什么异同？