

刘建平Pinard

十年研发，对数学统计学，数据挖掘，机器学习，大数据平台，大数据平台应用开发，大数据可视化感兴趣。

博客园 首页 新随笔 联系 订阅 管理

公告

★珠江追梦，饮岭南茶，恋鄂北家★
昵称：刘建平Pinard
园龄：1年4个月
粉丝：1011
关注：13
+加关注

< 2018年3月 >

日	一	二	三	四	五	六
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类(101)

0040. 数学统计学(4)
0081. 机器学习(62)
0082. 深度学习(10)
0083. 自然语言处理(23)
0121. 大数据挖掘(1)
0122. 大数据平台(1)
0123. 大数据可视化

随笔档案(101)

2017年8月(1)
2017年7月(3)
2017年6月(8)
2017年5月(7)
2017年4月(5)
2017年3月(10)
2017年2月(7)
2017年1月(13)
2016年12月(17)
2016年11月(22)
2016年10月(8)

常去的机器学习网站

52 NLP
Analytics Vidhya

scikit-learn 支持向量机算法库使用小结

之前通过一个系列对支持向量机(以下简称SVM)算法的原理做了一个总结，本文从实践的角度对scikit-learn SVM算法库的使用做一个小结。scikit-learn SVM算法库封装了libsvm 和 liblinear 的实现，仅仅重写了算法了接口部分。

1. scikit-learn SVM算法库使用概述

scikit-learn中SVM的算法库分为两类，一类是分类的算法库，包括SVC， NuSVC，和LinearSVC 3个类。另一类是回归算法库，包括SVR， NuSVR，和LinearSVR 3个类。相关的类都包裹在sklearn.svm模块之中。

对于SVC， NuSVC，和LinearSVC 3个分类的类， SVC和 NuSVC差不多，区别仅仅在于对损失的度量方式不同，而LinearSVC从名字就可以看出，他是线性分类，也就是不支持各种低维到高维的核函数，仅仅支持线性核函数，对线性不可分的数据不能使用。

同样的，对于SVR， NuSVR，和LinearSVR 3个回归的类， SVR和NuSVR差不多，区别也仅仅在于对损失的度量方式不同。LinearSVR是线性回归，只能使用线性核函数。

我们使用这些类的时候，如果有经验知道数据是线性可以拟合的，那么使用LinearSVC去分类 或者LinearSVR去回归，它们不需要我们去慢慢的调参去选择各种核函数以及对应参数，速度也快。如果我们对数据分布没有什么经验，一般使用SVC去分类或者SVR去回归，这就需要我们选择核函数以及对核函数调参了。

什么特殊场景需要使用NuSVC分类 和 NuSVR 回归呢？如果我们对训练集训练的错误率或者说支持向量的百分比有要求的时候，可以选择NuSVC分类 和 NuSVR。它们有一个参数来控制这个百分比。

这些类的详细使用方法我们在下面再详细讲述。

2. 回顾SVM分类算法和回归算法

我们先简要回顾下SVM分类算法和回归算法，因为这里面有些参数对应于算法库的参数，如果不先复习下，下面对参数的讲述可能会有些难以理解。

对于SVM分类算法，其原始形式是：

$$\begin{aligned} \min \quad & \frac{1}{2} ||w||_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y_i(w \bullet \phi(x_i) + b) \geq 1 - \xi_i \quad (i = 1, 2, \dots, m) \\ & \xi_i \geq 0 \quad (i = 1, 2, \dots, m) \end{aligned}$$

其中m为样本个数，我们的样本为 $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ 。 w, b 是我们的分离超平面的 $w \bullet \phi(x_i) + b = 0$ 系数， ξ_i 为第i个样本的松弛系数， C 为惩罚系数。 $\phi(x_i)$ 为低维到高维的映射函数。

通过拉格朗日函数以及对偶化后的形式为：

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1, j=1}^m \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^m \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \end{aligned}$$

其中和原始形式不同的 α 为拉格朗日系数向量。 $K(x_i, x_j)$ 为我们要使用的核函数。

对于SVM回归算法，其原始形式是：

$$\begin{aligned} \min \quad & \frac{1}{2} ||w||_2^2 + C \sum_{i=1}^m (\xi_i^{\vee} + \xi_i^{\wedge}) \\ \text{s.t.} \quad & -\epsilon - \xi_i^{\vee} \leq y_i - w \bullet \phi(x_i) - b \leq \epsilon + \xi_i^{\wedge} \\ & \xi_i^{\vee} \geq 0, \quad \xi_i^{\wedge} \geq 0 \quad (i = 1, 2, \dots, m) \end{aligned}$$

其中 m 为样本个数，我们的样本为 $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ 。 w, b 是我们的回归超平面的 $w \bullet x_i + b = 0$ 系数， ξ_i^\vee, ξ_i^\wedge 为第 i 个样本的松弛系数， C 为惩罚系数， ϵ 为损失边界，到超平面距离小于 ϵ 的训练集的点没有损失。 $\phi(x_i)$ 为低维到高维的映射函数。

通过拉格朗日函数以及对偶化后的形式为：

$$\underbrace{\min_{\alpha_i^\vee, \alpha_i^\wedge}} \frac{1}{2} \sum_{i=1, j=1}^m (\alpha_i^\wedge - \alpha_i^\vee)(\alpha_j^\wedge - \alpha_j^\vee) K(x_i, x_j) - \sum_{i=1}^m (\epsilon - y_i) \alpha_i^\wedge + (\epsilon + y_i) \alpha_i^\vee$$

$$s.t. \sum_{i=1}^m (\alpha_i^\wedge - \alpha_i^\vee) = 0$$

$$0 < \alpha_i^\vee < C \quad (i = 1, 2, \dots, m)$$

$$0 < \alpha_i^\wedge < C \quad (i = 1, 2, \dots, m)$$

其中和原始形式不同的 $\alpha^\vee, \alpha^\wedge$ 为拉格朗日系数向量。 $K(x_i, x_j)$ 为我们要使用的核函数。

3. SVM核函数概述

在scikit-learn中，内置的核函数一共有4种，当然如果你认为线性核函数不算核函数的话，那就只有三种。

- 1) 线性核函数 (Linear Kernel) 表达式为： $K(x, z) = x \bullet z$ ，就是普通的内积，LinearSVC 和 LinearSVR 只能使用它。
- 2) 多项式核函数 (Polynomial Kernel) 是线性不可分SVM常用的核函数之一，表达式为： $K(x, z) = (\gamma x \bullet z + r)^d$ ，其中， γ, r, d 都需要自己调参定义,比较麻烦。
- 3) 高斯核函数 (Gaussian Kernel)，在SVM中也称为径向基核函数 (Radial Basis Function,RBF)，它是libsvm默认核函数，当然也是scikit-learn默认的核函数。表达式为： $K(x, z) = \exp(-\gamma ||x - z||^2)$ ，其中， γ 大于0，需要自己调参定义。
- 4) Sigmoid核函数 (Sigmoid Kernel) 也是线性不可分SVM常用的核函数之一，表达式为： $K(x, z) = \tanh(\gamma x \bullet z + r)$ ，其中， γ, r 都需要自己调参定义。

一般情况下，对非线性数据使用默认的高斯核函数会有比较好的效果，如果你不是SVM调参高手的话，建议使用高斯核来做数据分析。

4. SVM分类算法库参数小结

这里我们对SVM分类算法库的重要参数做一个详细的解释，重点讲述调参的一些注意点。

参数	LinearSVC	SVC	NuSVC
惩罚系数C	即为我们第二节中SVM分类模型原型形式和对偶形式中的惩罚系数C，默认为1，一般需要通过交叉验证来选择一个合适的C。一般来说，如果噪音点较多时，C需要小一些。		NuSVC没有这个参数, 它通过另一个参数nu来控制训练集训练的误差率，等价于选择了一个C，让训练集训练后满足一个确定的误差率
nu	LinearSVC 和SVC没有这个参数，LinearSVC 和SVC使用惩罚系数C来控制惩罚力度。		nu代表训练集训练的误差率的上限，或者说支持向量的百分比下限，取值范围为(0,1],默认是0.5.它和惩罚系数C类似，都可以控制惩罚的力度。
核函数 kernel	LinearSVC没有这个参数，LinearSVC限制了只能使用线性核函数	核函数有四种内置选择，第三节已经讲到：'linear'即线性核函数，'poly'即多项式核函数，'rbf'即高斯核函数，'sigmoid'即sigmoid核函数。如果选择了这些核函数，对应的核函数参数在后面有单独的参数需要调。默认是高斯核'rbf'。 还有一种选择为"precomputed",即我们预先计算出所有的训练集和测试集的样本对应的Gram矩阵，这样 $K(x, z)$ 直接在对应的Gram矩阵中找到对应的位置的值得。 当然我们也可以自定义核函数，由于我没有用过自定义核函数，这里就不多讲了。	
正则化参数 penalty	仅仅对线性拟合有意义，可以选择'l1'即L1正则化 或者 'l2'即L2正则化。默认是L2正则化，如果我们需要产生稀疏话的系数的时候，可以选L1正则化,这和线性回归里面的Lasso回归类似。		SVC和NuSVC没有这个参数
是否用对偶形式	这是一个布尔变量，控制是否		SVC和NuSVC没有这个参数

机器学习库

机器学习路线图

深度学习进阶书

深度学习入门书

积分与排名

积分 - 295919

排名 - 617

阅读排行榜

1. 梯度下降 (Gradient Descent) 小结(90387)
2. 梯度提升树(GBDT)原理小结(42217)
3. 线性判别分析LDA原理总结(28887)
4. scikit-learn决策树算法类库使用小结(26671)
5. 谱聚类 (spectral clustering) 原理总结(19579)

评论排行榜

1. 梯度提升树(GBDT)原理小结(65)
2. 谱聚类 (spectral clustering) 原理总结(62)
3. 梯度下降 (Gradient Descent) 小结(55)
4. 卷积神经网络(CNN)反向传播算法(54)
5. 集成学习之Adaboost算法原理小结(49)

推荐排行榜

1. 梯度下降 (Gradient Descent) 小结(36)
2. 集成学习原理小结(14)
3. 集成学习之Adaboost算法原理小结(13)
4. 卷积神经网络(CNN)反向传播算法(12)
5. 协同过滤推荐算法总结(11)

优化dual	使用对偶形式来优化算法，默认是True,即采用上面第二节的分类算法对偶形式来优化算法。如果我们的样本量比特征数多，此时采用对偶形式计算量较大，推荐dual设置为False，即采用原始形式优化	
核函数参数 degree	LinearSVC没有这个参数，LinearSVC限制了只能使用线性核函数	如果我们在kernel参数使用了多项式核函数 'poly'，那么我们就需要对这个参数进行调参。这个参数对应 $K(x, z) = (\gamma x \bullet z + r)^d$ 中的 d 。默认是3。一般需要通过交叉验证选择一组合适的 γ, r, d
核函数参数 gamma	LinearSVC没有这个参数，LinearSVC限制了只能使用线性核函数	<p>如果我们在kernel参数使用了多项式核函数 'poly'，高斯核函数 'rbf'，或者sigmoid核函数，那么我们就需要对这个参数进行调参。</p> <p>多项式核函数中这个参数对应$K(x, z) = (\gamma x \bullet z + r)^d$中的$\gamma$。一般需要通过交叉验证选择一组合适的$\gamma, r, d$</p> <p>高斯核函数中这个参数对应$K(x, z) = \exp(-\gamma \ x - z\ ^2)$中的$\gamma$。一般需要通过交叉验证选择合适的$\gamma$</p> <p>sigmoid核函数中这个参数对应$K(x, z) = \tanh(\gamma x \bullet z + r)$中的$\gamma$。一般需要通过交叉验证选择一组合适的$\gamma, r$</p> <p>$\gamma$默认为'auto',即$\frac{1}{\text{特征维度}}$</p>
核函数参数 coef0	LinearSVC没有这个参数，LinearSVC限制了只能使用线性核函数	<p>如果我们在kernel参数使用了多项式核函数 'poly'，或者sigmoid核函数，那么我们就需要对这个参数进行调参。</p> <p>多项式核函数中这个参数对应$K(x, z) = (\gamma x \bullet z + r)^d$中的$r$。一般需要通过交叉验证选择一组合适的$\gamma, r, d$</p> <p>sigmoid核函数中这个参数对应$K(x, z) = \tanh(\gamma x \bullet z + r)$中的$r$。一般需要通过交叉验证选择一组合适的$\gamma, r$</p> <p>coef0默认为0</p>
样本权重 class_weight	指定样本各类别的权重，主要是为了防止训练集某些类别的样本过多，导致训练的决策过于偏向这些类别。这里可以自己指定各个样本的权重，或者用"balanced"，如果使用"balanced"，则算法会自己计算权重，样本量少的类别所对应的样本权重会高。当然，如果你的样本类别分布没有明显的偏倚，则可以不管这个参数，选择默认的"None"	
分类决策 decision_function_shape	LinearSVC没有这个参数，使用multi_class参数替代。	<p>可以选择'ovo'或者'ovr'。目前0.18版本默认是'ovo'。0.19版本将是'ovr'</p> <p>OvR(one vs rest)的思想很简单，无论你是多少元分类，我们都可以看做二元分类。具体做法是，对于第K类的分类决策，我们把所有第K类的样本作为正例，除了第K类样本以外的所有样本都作为负例，然后在上面做二元分类，得到第K类的分类模型。其他类的分类模型获得以此类推。</p> <p>OvO(one-vs-one)则是每次每次在所有的T类样本里面选择两类样本出来，不妨记为T1类和T2类，把所有的输出为T1和T2的样本放在一起，把T1作为正例，T2作为负例，进行二元分类，得到模型参数。我们一共需要T(T-1)/2次分类。</p> <p>从上面的描述可以看出OvR相对简单，但分类效果相对略差（这里指大多数样本分布情况，某些样本分布下OvR可能更好）。而OvO分类相对精确，但是分类速度没有OvR快。一般建议使用OvO以达到较好的分类效果。</p>
分类决策 multi_class	<p>可以选择 'ovr' 或者 'crammer_singer'</p> <p>'ovr'和SVC和nuSVC中的decision_function_shape对应的'ovr'类似。</p> <p>'crammer_singer'是一种改良版的'ovr'，说是改良，但是没有比'ovr'好，一般在应用中都不建议使用。</p>	SVC和nuSVC没有这个参数，使用decision_function_shape参数替代。
缓存大小 cache_size	LinearSVC计算量不大，因此	在大样本的时候，缓存大小会影响训练速度，因此如果机器内存大，推荐用500MB甚至1000MB。默认是200，即200MB。

不需要这个参数

5. SVM回归算法库参数小结

SVM回归算法库的重要参数巨大部分和分类算法库类似，因此这里重点讲述和分类算法库不同的部分，对于相同的部分可以参考上一节对应参数。

参数	LinearSVR	SVR	nuSVR
惩罚系数C	即为我们第二节中SVM分类模型原型形式和对偶形式中的惩罚系数C，默认为1，一般需要通过交叉验证来选择一个合适的C。一般来说，如果噪音点较多时，C需要小一些。大家可能注意到在分类模型里面，nuSVC使用了nu这个等价的参数控制错误率，就没有使用C，为什么我们nuSVR仍然有这个参数呢，不是重复了吗？这里的原因在回归模型里面，我们除了惩罚系数C还有一个距离误差 ϵ 来控制损失度量，因此仅仅一个nu不能等同于C.也就是说回归错误率是惩罚系数C和距离误差 ϵ 共同作用的结果。后面我们可以看到nuSVR中nu的作用。		
nu	LinearSVR 和SVR没有这个参数，用 ϵ 控制错误率		nu代表训练集训练的误差率的上限，或者说支持向量的百分比下限，取值范围为(0,1],默认是0.5. 通过选择不同的错误率可以得到不同的距离误差 ϵ 。也就是说这里的nu的使用和LinearSVR 和SVR的 ϵ 参数等价。
距离误差epsilon	即我们第二节回归模型中的 ϵ ，训练集中的样本需满足 $-\epsilon - \xi_i^V \leq y_i - w \bullet \phi(x_i) - b \leq \epsilon + \xi_i^A$		nuSVR没有这个参数，用nu控制错误率
是否用对偶形式优化dual	和SVC类似，可参考上一节的dual描述	SVR和NuSVR没有这个参数	
正则化参数penalty	和SVC类似，可参考上一节的penalty 描述	SVR和NuSVR没有这个参数	
核函数 kernel	LinearSVR没有这个参数，LinearSVR限制了只能使用线性核函数	和SVC, nuSVC类似，可参考上一节的kernel描述	
核函数参数degree, gamma 和coef0	LinearSVR没有这些参数，LinearSVR限制了只能使用线性核函数	和SVC, nuSVC类似，可参考上一节的kernel参数描述	
损失函数度量loss	<p>可以选择为'epsilon_insensitive'和'squared_epsilon_insensitive'，如果选择'epsilon_insensitive'，则损失度量满足$-\epsilon - \xi_i^V \leq y_i - w \bullet \phi(x_i) - b \leq \epsilon + \xi_i^A$二节的损失度量一样。是默认的SVM回归的损失度量标准形式。</p> <p>如果选择为'squared_epsilon_insensitive'，则损失度量满足$(y_i - w \bullet \phi(x_i) - b)^2 \leq \epsilon$，此时可见会少一个松弛系数。其优化过程我们在SVM原理系列里没有讲，但是目标函数优化过程是完全相似的。</p> <p>一般用默认的'epsilon_insensitive'就足够了。</p>	SVR和NuSVR没有这个参数	
缓存大小cache_size	LinearSVC计算量不大，因此不需要这个参数	在大样本的时候，缓存大小会影响训练速度，因此如果机器内存大，和SVC，nuSVC一样，推荐用500MB甚至1000MB。默认是200，即200MB。	

6. SVM算法库其他调参要点

上面已经对scikit-learn中类库的参数做了总结，这里对其他的调参要点做一个小结。

- 1) 一般推荐在做训练之前对数据进行归一化，当然测试集中的数据也需要归一化。。

- 2) 在特征数非常多的情况下，或者样本数远小于特征数的时候，使用线性核，效果已经很好，并且只需要选择惩罚系数C即可。
- 3) 在选择核函数时，如果线性拟合不好，一般推荐使用默认的高斯核'rbf'。这时我们主要需要对惩罚系数C和核函数参数 γ 进行艰苦的调参，通过多轮的交叉验证选择合适的惩罚系数C和核函数参数 γ 。
- 4) 理论上高斯核不会比线性核差，但是这个理论却建立在要花费更多的时间来调参上。所以实际上能用线性核解决问题我们尽量使用线性核。

(欢迎转载，转载请注明出处。欢迎沟通交流： pinard.liu@ericsson.com)

分类: [0081. 机器学习](#)

标签: [分类算法](#)

好文要顶

关注我

收藏该文

刘建平Pinard

关注 - 13

粉丝 - 1011

+加关注

« 上一篇：支持向量机原理(五)线性支持回归

» 下一篇：支持向量机高斯核调参小结

posted @ 2016-11-30 16:47 刘建平Pinard 阅读(7968) 评论(6) 编辑 收藏

评论列表

#1楼 2017-12-15 12:13 yueguangmanmande

在SVM回归中，样本权重class_weight怎么使用呢？在分类中，“指定样本各类别的权重，主要是为了防止训练集某些类别的样本过多”，在回归中的作用是什么呢？

支持(0)

反对(0)

#2楼[楼主] 2017-12-15 14:55 刘建平Pinard

@ yueguangmanmande
你好，回归里面没有class_weight，这里我复制的时候忘记删除，感谢指出错误，已经修改了。

支持(0)

反对(0)

#3楼 2018-01-11 11:23 Ariel.Xing

您好，我想请教一下，据您了解，sklearn的LinearSVC能否实现同时施加L1和L2正则化？如果可以，如何操作？万分感谢！

支持(0)

反对(0)

#4楼[楼主] 2018-01-11 11:30 刘建平Pinard

@ Ariel.Xing
你好，LinearSVC本身只能L1或者L2二选一，不能同时使用。
有些线性回归模型支持L1和L2一起优化，比如ElasticNet。

支持(0)

反对(0)

#5楼 2018-01-11 14:25 Ariel.Xing

@ 刘建平Pinard
非常感谢您的回复，我的确是想要将elastic net和SVM进行结合，但是我不确定自己的方法是否正确，我目前的思路如下，辛苦看一下是否合理，谢谢!
enet.set_params(alpha=alpha_optim)
coef_ = enet.fit(X, y).coef_
y_ = np.dot(X, coef_)
clf = svm.SVC()
clf.fit(X, y_)
result = clf.predict(X_test)

支持(0)

反对(0)

#6楼[楼主] 2018-01-12 12:22 刘建平Pinard

@ Ariel.Xing
个人觉得这样用比较牵强，难以有 好效果。你有没有比较过只用SVC的L2方法，和你上面的方法对比，哪个对测试集的预测准确率高？

支持(0)

反对(0)

刷新评论 刷新页面 返回顶部

https://www.cnblogs.com/pinard/p/6117515.html

5/6

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码：大型组态工控、电力仿真CAD与GIS源码库！

【推荐】云+校园计划邀请好友拼团有礼，奖励多多

【推荐】5分钟完成网站搭建 多种定制镜像，19.6元/月起

【活动】2050 科技公益大会 - 年青人因科技而团聚



最新IT新闻：

- 最后一博？ZUK、摩托罗拉之后，联想又要造机了
- 你愿意成为“身体黑客”吗？皮下植入芯片其实也痛苦
- 卡巴斯基：恶意软件Slingshot潜伏六年并通过路由器来传播
- HQ Trivia出现首位拿下所有奖金的玩家
- 谷歌上架两款拥有简洁体验的艺术和文化应用

» 更多新闻...



最新知识库文章：

- 写给自学者的入门指南
- 和程序员谈恋爱
- 学会学习
- 优秀技术人的管理陷阱
- 作为一个程序员，数学对你到底有多重要

» 更多知识库文章...

Copyright ©2018 刘建平Pinard