

应用场景

分类和预测是预测问题的两种主要类型，分类主要是预测分类标签（离散属性），例如“如何预测未来一段时间内哪些客户会流失”；而预测主要是建立连续值函数模型，预测给定自变量对应的因变量的数值（连续属性），例如“如何基于某产品的历史销售情况，以及节假日、气候和竞争对手等影响因素，对该产品销量进行预测”。在一些资料中，预测模型通常是跟分类模型放在一起介绍的，统称为分类与预测模型。为了便于总结、方便区分，我们将二者分开了。

回归分析和神经网络是预测模型中主要的两种方法，课程中涉及到的预测模型有 Lasso 模型、MLP 模型和 RBF 模型。

1. 预测模型在金融方面的应用

例如对于股票市场中股价涨跌幅度的预测。

为某支股票每天爬取数据一次，如下所示：

时间点	历史股价	交易特征					董监高持股特征				舆论情感特征	未来涨跌
		K_high	k_low	K_avg	涨跌	...	持股占比	变动比例	变动原因	...		
2019/6/17	4.13	4.16	4.06	4.1	-0.03	...	43.61%	1.62%	大宗交易	...	光头大阳线，介入	0.04
2019/6/18	4.10	4.18	4.07	4.14	0.04	...	43.61%	0	NaN	...	跳空下行，观望	-0.21
2019/6/19	4.14	3.97	3.9	3.93	-0.21	...	41.59%	-2.02%	竞价交易	...	跳空强势上扬，持有	0.09
2019/6/20	3.93	4.05	3.99	4.02	0.09	...	41.58%	-0.01%	二级市场买卖	...	强势上攻，寻机低吸	...
...
...	4.36	0.36
2019/9/17	4.36	4.81	4.34	4.72	0.36	...	49.26%	0.74%	镜中竞价	...	表现不佳，逢高减磅	

其中“涨跌”指标是由 K 线均值 K_avg 与历史股价之差而来。如 2019/6/17 时， $涨跌 = 4.1 - 4.13 = -0.03$ ，反映了当前股价比上一时间点股价的涨跌幅度的大小。在进行股票交易时，往往需要去预测下一个时间点的“涨跌”，这样方便在当前时间点决定该支股票是否买进或卖出，例如在 2019/9/16 时，假设我们能够预测到 2019/9/17 时将上涨 0.36，那么买入或者持仓可以获得更大收益等。

因此将当前的时间点的涨跌作为上一时间点的输出变量 y ，命名为“未来涨跌”。

特征向量 $X = (\text{历史股价、交易特征、董监高持股特征、舆论情感特征})$ 等，通过构建

预测模型，得到 2019/9/17 的“未来涨跌”（红色框），为股民进行股市操作提供参考。

根据未来时间点的股价对比当前股价的股价增幅或者增幅百分比作为标记做预测分析，是预测模型在金融方面的典型应用。

2. 预测模型在电商方面的应用

例如通过对店铺销售额的预测，提前制定营销策略以辅助店铺的发展。

某电商平台随机提取了 100 家同类商品的店铺月销售额数据，如下所示：

店铺ID	第1月销售额	第2月销售额	第3月销售额	...	第d月销售额	第d+1月销售额
001	99331	92460	148929	...	86757	138585
002	105036	143749	64459	...	104013	33771
003	98929	147673	25021	...	23839	95744
004	11857	43978	88419	...	123110	72528
...
100	51859	58388	145346	...	78933	43244

店铺的特征向量为 $X=(\text{第1月销售额、第2月销售额、...、第d月销售额})$ ，根据店铺往月的销售额去预测店铺的第 $d+1$ 月销售额 y ，此时表中最后一列将作为输出变量。在拟合得到预测模型后，对于同类商品的所有店铺而言，可以通过第 d 月之前的销售额得到第 $d+1$ 月销售额的预测值，在此基础上，进一步进行营销策略方面的决策。

3. 预测模型在教育方面的应用

例如通过预测学生成绩，找到成绩的影响因素，从而有针对性地对學生进行指导。

某在线学习平台记录了大量的学生学习行为数据。在某一时间节点下，随机从数据库中提取出 100 名学生的学习行为数据，如下所示：

学生ID	测试成绩	作业成绩	观看视频时长(min)	讨论发帖数	...	最终成绩
001	85	77	55	10	...	62
002	93	55	102	8	...	88
003	57	71	109	23	...	94
004	91	52	92	24	...	80
...
100	66	89	252	12	...	90

根据学生的特征向量 $X=(\text{测试成绩、作业成绩、观看视频时长、讨论发帖数})$ 等，

去预测学生的最终成绩 y ，根据得到的线性模型的特征系数可以找出与最终成绩相关的学习因素，从而在学生学习过程中抓住主要因素实时监测并给予必要的指导。

Lasso 模型

套索 (Lasso) 回归是回归分析中常用的线性模型，它实际上是一种改良的最小二乘法，是对线性回归进行 L_1 正则化以避免过拟合的模型。

1. 思想

- 在样本数据 N 中，样本特征向量为 X 由 d 个特征描述，即 $X=(x_1, x_2, \dots, x_d)$ ，每一个样本对应一个连续输出值 y 。

- 线性回归模型试图学得一个通过特征的线性组合来预测 y 值的函数，即

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots + \hat{\beta}_d x_d$$

- 属于线性模型，有监督学习。

注：关于线性回归详细介绍请查看“数据分析中级：金融数据、岗位分析/新三板：融资金额预测/数据建模与可视化/线性回归/2.1 线性回归模型”

2. 案例应用

1) 数据分析中级：金融数据、岗位分析案例二

对于“新三板”中的非上市股份有限公司，他们的主要目标是融资。融资是使用各种方式到金融市场上筹措或贷放资金的行为，一个公司的融资能力反映了该公司多方面的业务水平，也是银行等贷款方对公司的评估依据之一。在“数据分析中级：金融数据、岗位分析案例二”中，通过公司特征构建了 Lasso 模型去预测公司的融资金额，并挖掘公司融资金额背后的相关因素，从而为公司及银行等债权方提供预测支持。

该案例使用与中级案例一相同的 47 个公司特征进行建模分析，最终保留了 9 个特征，包括：具有高度相关性的“负债总计”和“资产总计”，一般正相关的“挂牌时长”、“注册资本”、“高管平均年龄”和“高管平均学历”，以及一般负相关的“营业收入增长率”、“总资产增长率”、“流动比率”，从而揭示了公司融资金额背后的相关因

素。

2) 数据分析中级：金融数据、岗位分析案例五

岗位薪资是青青招聘网站中非常重要的一个招聘信息特征，通过对岗位薪资的分析，一方面可以向招聘公司提供薪资制定的建议，另一方面可以基于应聘者的理想薪资，给出个人能力改进的指导。运营部门为了给用户在这方面提供更准确、更有效的服务，需要数据分析人员根据已有的公司行业、公司概况、岗位标签等招聘信息数据，构建岗位薪资预测模型（包括 Lasso 模型、MLP 模型和 RBF 模型），分析公司、岗位信息和岗位薪资的相关性。其中 Lasso 模型先后应用了两次。

第一次使用是为了挑选公司概况和岗位概述文本向量化后的最佳降维方案，只是一种简单的应用。使用降维后的公司概况和岗位概述作为输入特征，岗位薪资作为输出，创建 Lasso 模型，最终选择模型效果最好的降维方案。

第二次使用招聘信息中的 6 个特征（降维后的公司概况和岗位概述以及其他四个特征）创建 Lasso 模型预测岗位薪资，同时筛选出 23 个非零特征。从整体上来看：公司融资情况、公司员工数量和岗位薪资的相关性较弱；而学历要求越高，岗位薪资越高；转换后的公司概况、岗位概述字段也和岗位薪资有明显的相关性。虽然该案例最终选择了平均绝对误差（MAE）较低的 MLP 模型作为岗位薪资的预测模型，但 Lasso 模型也有效的探讨了岗位薪资背后的影响因素，为青青招聘网站的运营部门提供了参考。

3. 局限

1) 数据类型

- 只适用于 y 为连续值的情况
- X 中的连续变量应当接近标准正态分布

-
- X 中的多项式分布应当转为虚拟变量及多个二项分布
 - X 应当进行归一化以保证在同一尺度下，参数学习更加稳健
 - 适用于特征数量较多的场景，因为可以进行特征筛选
 - 每一个特征与 y 应当具有线性相关关系，如果是曲线相关关系则应该更细致抽取特征

取特征

注：

①判断一个特征与 y 是线性相关还是曲线相关关系，简单的方法就是画图，以特征为 x 轴，以 y 为 y 轴，观察散点的分布。

②更细致的抽取特征是指在业务层面增加数据特征，或者数据层面对某些复杂特征进行拆分等，最主要的还是前者。

2) 异常值

对异常值较为敏感，最好是剔除异常值后进行回归

3) 正则项

因为引入 L_1 正则项，非相关特征的系数更倾向于为 0，因此得到的线性模型更加稀疏。

4) 损失函数

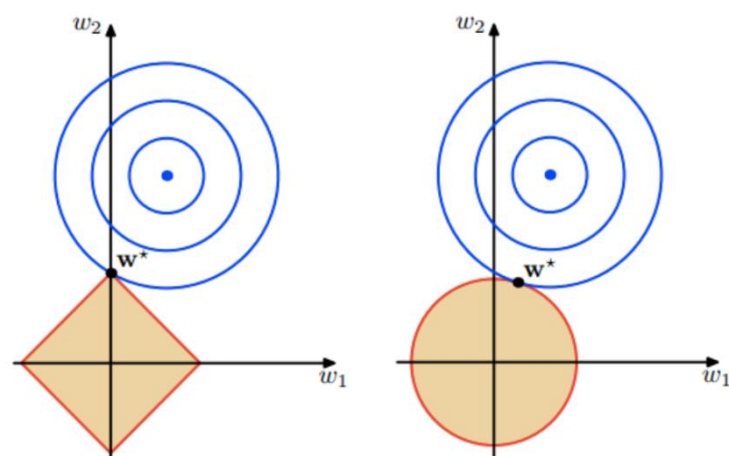
Lasso 模型的损失函数是平方损失函数（亦称均方误差），引入 L_1 正则化后的损失函数如下：

$$(y_i - \omega^T x_i)^2 + \lambda \|\omega\|_1$$

其中， x_i 是输入， ω 是参数向量， $\lambda \geq 0$ 是调整两项之间关系的系数， $\|\omega\|_1$ 表示参数向量 ω 的 L_1 范数。 L_1 范数和 L_2 范数正则化都有助于降低过拟合风险，但是 L_1 范数比后者更易于获得稀疏解，即求得的 ω 会有更少的非零分量。直观解释如下：

假定 ω 只有两个分量 ω_1, ω_2 ，将其作为两个坐标轴，绘制损失函数中第一项的“等值线”，即平方损失项取值相同的点的连线，再分别绘制 L_1 范数（下图左）和 L_2 范数（下

图右) 的等值线。



因为损失函数的解要在平方损失项与正则化项之间折中,所以解 w^* 出现在前后两项等值线的交点处。可以看到,采用 L_1 范数时,交点常出现在坐标轴上,使得 w_1 或 w_2 为 0;采用 L_2 范数时,交点常出现在某个象限中,使得 w_1 与 w_2 均非 0。具体的数学推导可以参见《机器学习》11.4 章节,或者链接 <https://www.cnblogs.com/heguanyou/archive/2017/09/23/7582578.html>。

注:关于损失函数与成本函数之间的关系,请查看本文档的扩展资料部分。

5) 超参数的选择

基于拟合优度系数 R 方(一般大于 0.6,拟合效果较好),以及特征稀疏度

6) 业务上的解读

- 获得非零特征后,根据系数的大小解读与 y 具有正/负相关的特征
- 模型仅能展示变量之间的相关关系,具体是否属于因果关系需要结合业务解释
- 由于是线性模型,因此筛选出来的非零特征与 y 具有比较强的线性相关性。被筛掉的特征并不一定就跟 y 没有关系,因为他们之间有可能是非线性关系。需要根据问题的实际背景对筛选结果进行分析。对于非线性关系的变量筛选方法有决策树、神经网络等。

7) 应用与调参

通过 sklearn 库中的类 `Lasso()` 实现 Lasso 模型，主要调节的参数有 α (正则项系数 λ) 与 tol (收敛阈值)。其中 α 越大，正则力度越强，对变量的选择能力越强，得到的结果也就更加稀疏； tol 越小，要求的收敛稳定性越高，需要的收敛时间也就越长。应用与调参的过程如下所示：

- 对 X 进行归一化
- 设置超参数 α 与 tol 的可能取值
- 通过网格搜索法确定最优超参数组合
- 基于测试集评价最优超参数下模型的效果

Lasso()参数

sklearn 中 Lasso()中共 11 个参数，如下所示：

序号	参数	中文名	默认取值	备注
1	alpha	正则项系数 λ	1.0	
2	fit_intercept	是否存在截距或偏差	True	
3	normalize	是否标准化	False	
4	precompute	预计算	False	
5	copy_X	复制 X	True	
6	max_iter	收敛最大迭代次数	1000	
7	tol	收敛阈值	0.0001	
8	warm_start	热启动	False	如果为 True，则使用上次的训练结果作为初始化参数，否则擦除上次的训练结果
9	positive	正系数	False	True 时强制系数为正
10	random_state	随机种子	None	
11	selection	系数选择	'cyclic'	'cyclic'/'random'

注：

1) normalize 参数

当 fit_intercept=False 时，忽略该参数。如果 True，则回归前对 X 进行归一化处理，如果 False，则拟合模型前 X 进行标准化处理。

2) precompute 参数

是否使用预计算的 Gram 矩阵来加速计算。对于稀疏输出这个选项永远为 True。

Gram 矩阵介绍可参考：

<https://blog.csdn.net/wangyang20170901/article/details/79037867>

3) selection 参数

若设为 'random'，每次迭代都会更新一个随机系数，而不是默认情况下按顺序遍历特性。设为 'random' 通常会更快的收敛，尤其是 tol 比 $1e-4$ 大的情况下。

4) 更多参考可查看官方文档：

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

损失函数与成本函数

损失函数 (loss function) 是定义在单个训练样本 (x_i, y_i) 上的损失, 即预测值 $f(x_i)$ 与实际值 y_i 之间的误差。常见的损失函数有平方损失函数、绝对损失函数、0-1 损失函数和 log 损失函数等。

成本函数 (cost function) 是定义在整体训练集上的, 即所有样本损失函数的平均值。

例如, 对于 Lasso 模型, 其损失函数是平方损失函数

$$L(f(x_i; \omega), y_i) = (y_i - \omega^T x_i)^2 ,$$

其成本函数是

$$J(\omega) = \frac{1}{N} \sum_{i=1}^N (y_i - \omega^T x_i)^2 ,$$

其中 N 是样本量。但是如果模型过于复杂会导致过拟合现象, 因此引入正则化项来评估模型的复杂程度, 从而求解最优化问题

$$\omega^* = \operatorname{argmin} \frac{1}{N} \sum_{i=1}^N (y_i - \omega^T x_i)^2 + \lambda \|\omega\|_1$$

Ridge 模型和 Elasticnet 模型

1. 岭 (Ridge) 回归

在数据集中特征数量不多、每一个都有重要作用时，往往会使用另一种引入 L_2 正则项的岭 (Ridge) 回归。它保留了全部特征，仅降低了特征变量系数值，不具有稀疏化的倾向。

注：Ridge 回归可通过 sklearn 中 `Ridge()` 方法实现，参考官方文档

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

2. 弹性网络 (Elasticnet) 回归

而在我们发现用 Lasso 回归太过 (太多特征被稀疏为 0)，而 Ridge 回归正则化不够 (回归系数衰减太慢) 的时候，可以考虑使用既引入 L_1 正则项又引入 L_2 正则项的弹性网络 (Elasticnet) 回归，得到比较好的结果。

注：Elasticnet 回归可通过 sklearn 中 `ElasticNet()` 方法实现，参考官方文档

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

MLP 模型

MLP(Multi-Layer Perceptron ,多层感知机)模型也被称为多层前馈神经网络(Multi-layer Feedforward Neural Networks) , 或者被泛称为神经网络 , 在自然语言处理上具有非常卓越的表现。同时在该模型基础上的 RNN (循环神经网络)、CNN (卷积神经网络) 等在目前图像识别、语音识别以及无人驾驶等领域均有非常广泛的应用。

1. 思想

- 在样本数据 N 中 , 样本特征向量为 X 由 d 个特征描述 , 即 $X=(x_1, x_2, \dots, x_d)$, 每一个样本对应一个连续输出值/类别标签 y 。
- MLP 模型以 M-P 神经元为单位 , 由输入层、隐藏层、输出层三部分组成 , 其中输入层负责接收样本特征 , 隐层与输出层神经元对信号进行加工 , 最终结果由输出层神经元输出。
- 可用于拟合或者分类 , 有监督学习。

注：关于 MLP 模型详细介绍请查看“数据分析中级：金融数据、岗位分析/招聘数据：岗位薪资预测/数据建模与可视化/2. MLP 模型/2.1MLP 算法中的介绍”

2. 案例应用

岗位薪资是青青招聘网站中非常重要的一个招聘信息特征 , 通过对岗位薪资的分析 , 一方面可以向招聘公司提供薪资制定的建议 , 另一方面可以基于应聘者的理想薪资 , 给出个人能力改进的指导。在“数据分析中级：金融数据、岗位分析案例五”中 , 运营部门为了给用户在这方面提供更准确、更有效的服务 , 需要数据分析人员根据已有的公司行业、公司概述、岗位标签等招聘信息数据 , 构建 MLP 岗位薪资预测模型。本案例中还使用了 RBF 模型和 Lasso 模型两种预测模型 , 但是最终选择了平均绝对误差 MAE

最低的 MLP 模型作为岗位薪资的预测模型，为青青招聘网站的运营部门提供了参考。

3. 局限

1) 数据特点

- 适用于特征数少于样本量的情况
- 数据同样需要归一化，否则激活函数收到的映射不统一

2) 异常值

MLP 的拟合能力较强，容易过拟合，训练可能会训练出异常值的结论，应当删除异常值后拟合

3) 损失函数

- 平方损失函数（亦称均方误差）加 L2 正则项
- MLP 模型可能收敛到局部最优，因此需要多次对权重进行随机初始化查看训练效果

4) 超参数的选择

- 通过 sklearn 库中的类 MLPRegressor()实现 MLP 模型，主要超参数包括 hidden_layer_sizes（隐藏层节点数列表）、activation（激活函数）和 solver（优化算法）
- 基于 R 方评价模型，选择超参数的最佳取值

5) 可解释性

MLP 模型不具有很强的可解释性，也可用于分类，但是无法判断不同特征的贡献。

6) 应用与调参

- 对 X 进行归一化

-
- 设置超参数 `hidden_layer_sizes`, `activation`, `solver` 等的可能取值(建议优先尝试结构简单的模型)
 - 通过网格搜索法确定最优超参数组合
 - 基于测试集评价最优超参数下模型的效果

MLPRegressor()参数

sklearn 中 MLPRegressor()中共 14 个参数，如下所示：

序号	参数	中文名	默认取值	备注
1	hidden_layer_sizes	隐藏层节点数列表	100	
2	activation	激活函数	'relu'	可选：'identity'、 'logistic'、'tanh'、'relu'
3	solver	优化算法	'adam'	可选：'lbfgs'、'sgd'、 'adam'
4	alpha	L2 正则化参数	0.0001	
5	batch_size	随机优化器的小批量尺寸	'auto'	
6	learning_rate	学习率	'constant'	可选：'constant'、 'invscaling'、'adaptive'
7	learning_rate_init	学习率初始值	0.001	仅'sgd'、'adam'时可用
8	power_t	逆缩放学习率指数	0.5	仅'invscaling'时可用
9	max_iter	最大迭代次数	200	
10	shuffle	迭代中是否洗牌	True	仅'sgd'、'adam'时可用
11	random_state	随机种子	None	
12	tol	收敛阈值	0.0001	
13	verbose	启用详细输出	False	
14	warm_start	热启动	False	
15	momentum	梯度下降更新量	0.9	区间[0, 1];仅'sgd'时可用
16	nesterovs_momentum	nesterovs 动量	True	仅'sgd' 和 momentum>0 时可用
17	early_stopping	启用提前停止	False	
18	validation_fraction	预留验证集比例	0.1	区间[0, 1];仅'early_stopping' 启用时可用
19	beta_1	β_1	0.9	区间[0, 1];仅'adam'时可用
20	beta_2	β_2	0.999	区间[0, 1];仅'adam'时可用
21	epsilon	ϵ	1e-08	仅'adam'时可用
22	n_iter_no_change	无改变迭代次数	10	不满足 tol 改进的最大纪元数

注：

1) activation 参数

- 'identity' 无操作激活： $f(x)=x$
- 'logistic' sigmoid 函数： $f(x)=1/(1+\exp(-x))$
- 'tanh' 双曲线 tan 函数： $f(x)=\tanh(x)$
- 'relu' 修正线性单元函数： $f(x) = \max(0, x)$

2) solver 参数

- 'lbfgs' 拟牛顿优化算法中的一种，是基于 BFGS 算法的一种近似
- 'sgd' 随机梯度下降。可参考人工智能中级：数字识别、人脸识别/AI 课程综述/神经网络基础中的介绍

- 'adam' 自适应矩估计：是一种可以替代 SGD 过程的一阶优化算法，他能基于训练数据迭代地更新神经网络权重

其中 'adam' 在相对较大的数据集 (包含数千个或更多的训练样本) 上, 训练时间和验证分数方面都表现较好。而对于小数据集, 'lbfgs' 可以更快地收敛并执行得更好。关于诸多优化算法的比较可参考:

<https://blog.csdn.net/u010089444/article/details/76725843>

<https://blog.csdn.net/u012759136/article/details/52302426>

http://blog.sina.com.cn/s/blog_eb3aea990101gflj.html

3) batch_size 参数

当优化器选择 'lbfgs', 则不考虑小批量。默认 'auto' 即小批量尺寸为 (200, n_sample)。

4) learning_rate 参数

- 'constant': 表示根据超参数 'learning_rate_init' 给出恒定学习率
- 'invscaling': 使用超参数 'power_t' 的逆缩放指数在每个时间步 't' 逐渐降低的学习速率, $\text{effective_learning_rate} = \text{learning_rate_init} / \text{pow}(t, \text{power_t})$
- 'adaptive' 自适应: 在损失值持续减少的情况下, 学习率将保持超参数 'learning_rate_init' 不变。当两个连续的时间间隔不能减少至少 tol 的损失值, 或者如果 'early_stop' 打开, 则验证分数不能增加至少 tol, 则当前学习率除以 5 仅当优化器选择 'sgd' 时有效。

5) momentum 参数和 nesterovs_momentum 参数

关于动量介绍可参考: <https://blog.csdn.net/u012328159/article/details/80311892>

6) early_stopping 参数

自动留出 10% 的训练数据作为验证, 并在至少两个连续时期验证得分没有提高的 tol 时终止训练。

仅当优化器选择 'sgd' 或 'adam' 时有效。

7) beta_1、beta_2 和 epsilon 参数

adam 算法中的参数, 可参考 adam 算法介绍: <https://www.jianshu.com/p/3e363f5e1a79>

8) 更多参考可查看官方文档:

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

RBF 模型

RBF (Radical Basis Function, 径向基函数) 可以处理系统内的难以解析的规律性, 具有良好的泛化能力, 并有很快的学习收敛速度, 已成功应用于非线性函数逼近、时间序列分析、数据分类、模式识别、信息处理、图像处理、系统建模、控制和故障诊断等。

1. 思想

- 在样本数据 N 中, 样本特征向量为 X 由 d 个特征描述, 即 $X=(x_1, x_2, \dots, x_d)$, 每一个样本对应一个连续输出值 y
- RBF 模型通过构建一个单隐层前馈神经网络, 使用径向基函数作为隐层神经元激活函数, 输出层则是对隐层神经元输出的线性组合
- 对比 SVM, RBF 神经网络隐含层是对输入点做了聚类, 计算样本与聚类中心的距离, 而 SVM 中的 RBF 核函数可以看做是与每一个输入点的距离
- 可用于拟合, 有监督学习

注:

①关于 RBF 神经网络详细介绍请查看“数据分析中级: 金融数据、岗位分析/招聘数据: 岗位薪资预测/数据建模与可视化/3. RBF 模型/3.1 RBF 算法的介绍”

②关于径向基函数 RBF 与 RBF 核的对比可参考:

<https://blog.csdn.net/u013630349/article/details/48162589>

2. 案例应用

岗位薪资是青青招聘网站中非常重要的一个招聘信息特征, 通过对岗位薪资的分析, 一方面可以向招聘公司提供薪资制定的建议, 另一方面可以基于应聘者的理想薪资, 给出个人能力改进的指导。在“数据分析中级: 金融数据、岗位分析案例五”中, 运营部门为了给用户在这方面提供更准确、更有效的服务, 需要数据分析人员根据已有的公

司行业、公司概述、岗位标签等招聘信息数据，构建 RBF 岗位薪资预测模型。本案例中还使用了 MLP 模型和 Lasso 模型两种预测模型，根据测试集的 MAE（平均绝对误差），发现 RBF 模型的 MAE 远高出其他两种模型，效果较差，最终选择了 MLP 模型作为岗位薪资的预测模型，为青青招聘网站的运营部门提供了参考。

3. 局限

1) 数据分布

RBF 神经网络对数据分布的要求与 SVM 对数据分布的要求相似，因为 RBF 神经网络需要对每个样本点与聚类中心计算 RBF 函数

2) 损失函数

平方损失函数（亦称均方误差）

3) 超参数的选择

- 目前还没有可直接调用的 RBF 模型，只要求掌握给出的 RBF 模型代码的使用即可。该模型需要选择的超参数只有 numNodes（隐层节点数目）
- 基于 MAE 评价模型，确定超参数的最佳取值
- 对比 MLP，RBF 神经网络仅有一层隐藏层，不需要调整隐藏层参数

4) 可解释性

RBF 模型不具有很强的可解释性，可以用于分类，但是无法判断不同特征的贡献。

5) 应用与调参

- 对 X 进行归一化
- 设置隐层节点数的可能取值
- 通过均方误差确定最优超参数

-
- 基于测试集评价最优超参数下模型的效果

4. 拓展资料

对 RBF 模型感兴趣的同学可查看链接：

<https://www.cnblogs.com/zhangchaoyang/articles/2591663.html>

<https://blog.csdn.net/zouxy09/article/details/13297881>