CSDN
不 止 于 代 码

# 3.3 Scikit learn中的Logistic回归实现

CSDN学院

2017年10月

AI 001

# 大纲



- Logistic回归基本原理
- 多类Logistic回归
- Scikit learn 中的Logistic回归实现
- 案例分析
- 分类模型的评价
- 模型选择与参数调优

# Scikit learn 中的LogisticRegression实现



- Scikit learn提供的LogisticRegression实现为：
  LogisticRegression(*penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='liblinear', max_iter=100, multi_class='ovr', verbose=0, warm_start=False, n_jobs=1*)
  - Logistic回归的正则参数：*penalty*、*C*
  - 优化求解参数： *dual*、*solver*、*max_iter*、*tol*、*warm_start*
  - 模型参数： *multi_class*、*fit_intercept*、*intercept_scaling*、*class_weight*



http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

# LogisticRegression参数列表



| 参数 | 说明 |
| --- | --- |
| penalty | 惩罚函数 / 正则函数，支持L2正则和L1正则，缺省：L2 |
| dual | 原问题（primal）还是对偶问题求解。对偶只支持L2正则和liblinear solver。当样本数n_samples>特征数目n_features时，缺省：False |
| tol | 迭代终止判据的误差范围。 缺省:1e-4 |
| C | C=1/$\lambda$，缺省：1 |
| fit_intercept | 是否在决策函数中加入截距项。如果数据已经中心化，可以不用。缺省：True |
| intercept_scaling | 截距缩放因子，当fit_intercept为True且liblinear solver有效<br>所以还是对y做标准化预处理 |
| class_weight | 不同类别样本的权重，用户指定每类样本权重或‘balanced’（每类样本权重与该类样本出现比例成反比）。缺省：None |
| random_state | 混合数据的伪随机数。缺省：None |

# LogisticRegression参数列表

| 参数 | 说明 |
| --- | --- |
| solver | 优化求解算法，可为‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’。缺省：liblinear |
| max_iter | 最大迭代次数，当newton-cg, sag and lbfgs solvers时有效。缺省：100 |
| multi_class | 多类分类处理策略，可为‘ovr’, ‘multinomial’。‘ovr’为1对多，将多类分类转化为多个两类分类问题，multinomial为softmax分类。缺省：‘ovr’ |
| verbose | 是否详细输出 |
| warm_start | 是否热启动（用之前的结果作为初始化），对liblinear solver无效。缺省：False |
| n_jobs | 多线程控制。缺省值-1，算法自动检测可用CPU核，并使用全部核 |

# 多类分类任务



- multi_class参数决定了多类分类的实现方式
- ‘ovr’：即1对其他（one-vs-rest，OvR），将多类分类转化为多个二类分类任务。为了完成第*c*类的分类决策，将所有第*c*类的样本作为正例，除了第*c*类样本以外的所有样本都作为负例。
- ‘multinomial’：多对多（many-vs-many，MvM），即softmax回归模型。
- OvR相对简单，但分类效果相对略差
  - 大多数情况，不排除某些情况下OvR更好
- MvM分类相对精确，但分类速度较OvR慢
- multi_class选择会影响优化算法solver参数的选择
  - OvR：可用所有的slover
  - Multinomial： 只能选择newton-cg, lbfgs和sag / saga

# 优化求解算法solver



http://www.csie.ntu.edu.tw/~cjlin/liblinear/

- liblinear：使用了开源的liblinear库实现，使用坐标轴下降法来迭代优化损失函数
- sag：随机平均梯度下降（Stochastic Average Gradient），是梯度下降法的变种，每次迭代仅用一部分的样本来计算梯度，适合于样本多的情况
- saga： sag的增强版本
- lbfgs：拟牛顿法的一种，利用损失函数二阶导数矩阵（Hessian矩阵）来迭代优化损失函数
- newton-cg：牛顿法家族的一种（ 共轭梯度 ）

# ► 优化求解算法solver



- 对小数据集，‘liblinear’ 是一个很好的选择，而 ‘sag’ 和‘saga’ 对大数据集更快。
- 对多类分类问题，只有 ‘newton-cg’, ‘sag’, ‘saga’ 和‘lbfgs’支持MvM（multinomial），‘liblinear’ 只支持OvR（one-versus-rest）的方式。
- ‘newton-cg’, ‘lbfgs’ 和‘sag’ 支持L2正则，而 ‘liblinear’ 和‘saga’ 支持 L1 正则。

- 注意：‘sag’ 和‘saga’ 只有当特征有类似的尺度（scale）时能保证快速收敛。（对数据做标准化预处理）

# 优化求解算法solver选择



| 正则 | 求解算法 | 应用场景 |
| --- | --- | --- |
| L1 | liblinear | 如果模型的特征非常多，希望一些不重要的特征系数归零从而让模型系数稀疏的话，可以使用L1正则化。liblinear适用于小数据集 |
| L1 | saga | 当数据量较大，且选择L1，只能采用saga |
| L2 | liblinear | libniear只支持多元逻辑回归的OvR，不支持多项分布损失（MvM），但MVM相对精确。 |
| L2 | lbfgs/newton-cg/sag | 较大数据集，支持OvR和 MvM两种多元logit回归。 |
| L2 | sag / saga | 如果样本量非常大，sag / sga是第一选择 |

- 对于大数据集，可以考虑使用SGDClassifier，并使用logloss。

# 类别权重class_weight



- class_weight用于不同类别样本数目不均衡的情况
- 另开一小节讲解

CSDN
不 止 于 代 码

# THANK YOU


AI 100