

# Python机器学习算法



[Python机器学习算法\\_下载链接1](#)

著者:赵志勇

出版者:电子工业出版社

出版时间:2017-7

装帧:平装

isbn:9787121313196

《Python机器学习算法》是一本机器学习入门读物，注重理论与实践的结合。全书主要包括6个部分，每个部分均以典型的机器学习算法为例，从算法原理出发，由浅入深，详细介绍算法的理论，并配合目前流行的Python语言，从零开始，实现每一个算法。

，以加强对机器学习算法理论的理解、增强实际的算法实践能力，最终达到熟练掌握每一个算法的目的。与其他机器学习类图书相比，《Python机器学习算法》同时包含算法理论的介绍和算法的实践，以理论支撑实践，同时，又将复杂、枯燥的理论用简单易懂的形式表达出来，促进对理论的理解。

作者介绍:

目录: 0 绪论	1
0.1 机器学习基础	1
0.1.1 机器学习的概念	1
0.1.2 机器学习算法的分类	2
0.2 监督学习	3
0.2.1 监督学习	3
0.2.2 监督学习的流程	3
0.2.3 监督学习算法	4
0.3 无监督学习	4
0.3.1 无监督学习	4
0.3.2 无监督学习的流程	4
0.3.3 无监督学习算法	5
0.4 推荐系统和深度学习	6
0.4.1 推荐系统	6
0.4.2 深度学习	6
0.5 Python和机器学习算法实践	6
参考文献	7
第一部分 分类算法	
1 Logistic Regression	10
1.1 Logistic Regression模型	10
1.1.1 线性可分VS线性不可分	10
1.1.2 Logistic Regression模型	11
1.1.3 损失函数	13
1.2 梯度下降法	14
1.2.1 梯度下降法的流程	14
1.2.2 凸优化与非凸优化	15
1.2.3 利用梯度下降法训练Logistic Regression模型	17
1.3 梯度下降法的若干问题	18
1.3.1 选择下降的方向	18
1.3.2 步长的选择	19
1.4 Logistic Regression算法实践	20
1.4.1 利用训练样本训练Logistic Regression模型	20
1.4.2 最终的训练效果	22
1.4.3 对新数据进行预测	23
参考文献	26
2 Softmax Regression	27
2.1 多分类问题	27
2.2 Softmax Regression算法模型	28
2.2.1 Softmax Regression模型	28
2.2.2 Softmax Regression算法的代价函数	28
2.3 Softmax Regression算法的求解	29
2.4 Softmax Regression与Logistic Regression的关系	31
2.4.1 Softmax Regression中的参数特点	31
2.4.2 由Softmax Regression到Logistic Regression	31
2.5 Softmax Regression算法实践	32

2.5.1 对Softmax Regression算法的模型进行训练	33
2.5.2 最终的模型	34
2.5.3 对新的数据的预测	35
参考文献	39
3 Factorization Machine	40
3.1 Logistic Regression算法的不足	40
3.2 因子分解机FM的模型	42
3.2.1 因子分解机FM模型	42
3.2.2 因子分解机FM可以处理的问题	43
3.2.3 二分类因子分解机FM算法的损失函数	43
3.3 FM算法中交叉项的处理	43
3.3.1 交叉项系数	43
3.3.2 模型的求解	44
3.4 FM算法的求解	45
3.4.1 随机梯度下降 (Stochastic Gradient Descent)	45
3.4.2 基于随机梯度的方式求解	45
3.4.3 FM算法流程	46
3.5 因子分解机FM算法实践	49
3.5.1 训练FM模型	50
3.5.2 最终的训练效果	53
3.5.3 对新的数据进行预测	55
参考文献	57
4 支持向量机	58
4.1 二分类问题	58
4.1.1 二分类的分隔超平面	58
4.1.2 感知机算法	59
4.1.3 感知机算法存在的问题	61
4.2 函数间隔和几何间隔	61
4.2.1 函数间隔	62
4.2.2 几何间隔	62
4.3 支持向量机	63
4.3.1 间隔最大化	63
4.3.2 支持向量和间隔边界	64
4.3.3 线性支持向量机	65
4.4 支持向量机的训练	66
4.4.1 学习的对偶算法	66
4.4.2 由线性支持向量机到非线性支持向量机	68
4.4.3 序列最小最优化算法SMO	69
4.5 支持向量机SVM算法实践	74
4.5.1 训练SVM模型	74
4.5.2 利用训练样本训练SVM模型	81
4.5.3 利用训练好的SVM模型对新数据进行预测	85
参考文献	88
5 随机森林	89
5.1 决策树分类器	89
5.1.1 决策树的基本概念	89
5.1.2 选择最佳划分的标准	91
5.1.3 停止划分的标准	94
5.2 CART分类树算法	95
5.2.1 CART分类树算法的基本原理	95
5.2.2 CART分类树的构建	95
5.2.3 利用构建好的分类树进行预测	98
5.3 集成学习 (Ensemble Learning)	99
5.3.1 集成学习的思想	99

5.3.2 集成学习中的典型方法	99
5.4 随机森林 (Random Forests)	101
5.4.1 随机森林算法模型	101
5.4.2 随机森林算法流程	102
5.5 随机森林RF算法实践	104
5.5.1 训练随机森林模型	105
5.5.2 最终的训练结果	109
5.5.3 对新数据的预测	110
参考文献	113
6 BP神经网络	114
6.1 神经元概述	114
6.1.1 神经元的基本结构	114
6.1.2 激活函数	115
6.2 神经网络模型	116
6.2.1 神经网络的结构	116
6.2.2 神经网络中的参数说明	117
6.2.3 神经网络的计算	117
6.3 神经网络中参数的求解	118
6.3.1 神经网络损失函数	118
6.3.2 损失函数的求解	119
6.3.3 BP神经网络的学习过程	120
6.4 BP神经网络中参数的设置	126
6.4.1 非线性变换	126
6.4.2 权重向量的初始化	126
6.4.3 学习率	127
6.4.4 隐含层节点的个数	127
6.5 BP神经网络算法实践	127
6.5.1 训练BP神经网络模型	128
6.5.2 最终的训练效果	132
6.5.3 对新数据的预测	133
参考文献	136
第二部分 回归算法	
7 线性回归	138
7.1 基本线性回归	138
7.1.1 线性回归的模型	138
7.1.2 线性回归模型的损失函数	139
7.2 线性回归的最小二乘解法	140
7.2.1 线性回归的最小二乘解法	140
7.2.2 广义逆的概念	141
7.3 牛顿法	141
7.3.1 基本牛顿法的原理	141
7.3.2 基本牛顿法的流程	142
7.3.3 全局牛顿法	142
7.3.4 Armijo搜索	144
7.3.5 利用全局牛顿法求解线性回归模型	145
7.4 利用线性回归进行预测	146
7.4.1 训练线性回归模型	147
7.4.2 最终的训练结果	149
7.4.3 对新数据的预测	150
7.5 局部加权线性回归	152
7.5.1 局部加权线性回归模型	152
7.5.2 局部加权线性回归的最终结果	153
参考文献	154
8 岭回归和Lasso回归	155

8.1 线性回归存在的问题	155
8.2 岭回归模型	156
8.2.1 岭回归模型	156
8.2.2 岭回归模型的求解	156
8.3 Lasso回归模型	157
8.4 拟牛顿法	158
8.4.1 拟牛顿法	158
8.4.2 BFGS校正公式的推导	158
8.4.3 BFGS校正的算法流程	159
8.5 L-BFGS求解岭回归模型	162
8.5.1 BFGS算法存在的问题	162
8.5.2 L-BFGS算法思路	162
8.6 岭回归对数据的预测	165
8.6.1 训练岭回归模型	166
8.6.2 最终的训练结果	168
8.6.3 利用岭回归模型预测新的数据	168
参考文献	171
9 CART树回归	172
9.1 复杂的回归问题	172
9.1.1 线性回归模型	172
9.1.2 局部加权线性回归	173
9.1.3 CART算法	174
9.2 CART回归树生成	175
9.2.1 CART回归树的划分	175
9.2.2 CART回归树的构建	177
9.3 CART回归树剪枝	179
9.3.1 前剪枝	179
9.3.2 后剪枝	180
9.4 CART回归树对数据预测	180
9.4.1 利用训练数据训练CART回归树模型	180
9.4.2 最终的训练结果	182
9.4.3 利用训练好的CART回归树模型对新的数据预测	185
参考文献	187
第三部分 聚类算法	
10 K-Means	190
10.1 相似性的度量	190
10.1.1 闵可夫斯基距离	191
10.1.2 曼哈顿距离	191
10.1.3 欧氏距离	191
10.2 K-Means算法原理	192
10.2.1 K-Means算法的基本原理	192
10.2.2 K-Means算法步骤	193
10.2.3 K-Means算法与矩阵分解	193
10.3 K-Means算法实践	195
10.3.1 导入数据	196
10.3.2 初始化聚类中心	197
10.3.3 聚类过程	198
10.3.4 最终的聚类结果	199
10.4 K-Means++算法	200
10.4.1 K-Means++算法存在的问题	200
10.4.2 K-Means++算法的基本思路	202
10.4.3 K-Means++算法的过程和最终效果	204
参考文献	205
11 Mean Shift	206

11.1 Mean Shift向量	206
11.2 核函数	207
11.3 Mean Shift算法原理	209
11.3.1 引入核函数的Mean Shift向量	209
11.3.2 Mean Shift算法的基本原理	210
11.4 Mean Shift算法的解释	212
11.4.1 概率密度梯度	212
11.4.2 Mean Shift向量的修正	213
11.4.3 Mean Shift算法流程	213
11.5 Mean Shift算法实践	217
11.5.1 Mean Shift的主过程	218
11.5.2 Mean Shift的最终聚类结果	219
参考文献	221
12 DBSCAN	222
12.1 基于密度的聚类	222
12.1.1 基于距离的聚类算法存在的问题	222
12.1.2 基于密度的聚类算法	225
12.2 DBSCAN算法原理	225
12.2.1 DBSCAN算法的基本概念	225
12.2.2 DBSCAN算法原理	227
12.2.3 DBSCAN算法流程	228
12.3 DBSCAN算法实践	231
12.3.1 DBSCAN算法的主要过程	232
12.3.2 Mean Shift的最终聚类结果	234
参考文献	236
13 Label Propagation	237
13.1 社区划分	237
13.1.1 社区以及社区划分	237
13.1.2 社区划分的算法	238
13.1.3 社区划分的评价标准	239
13.2 Label Propagation算法原理	239
13.2.1 Label Propagation算法的基本原理	239
13.2.2 标签传播	240
13.2.3 迭代的终止条件	242
13.3 Label Propagation算法过程	244
13.4 Label Propagation算法实践	244
13.4.1 导入数据	245
13.4.2 社区的划分	246
13.4.3 最终的结果	247
参考文献	248
第四部分 推荐算法	
14 协同过滤算法	250
14.1 推荐系统的概述	250
14.1.1 推荐系统	250
14.1.2 推荐问题的描述	251
14.1.3 推荐的常用方法	251
14.2 基于协同过滤的推荐	252
14.2.1 协同过滤算法概述	252
14.2.2 协同过滤算法的分类	252
14.3 相似度的度量方法	253
14.3.1 欧氏距离	254
14.3.2 皮尔逊相关系数 (Pearson Correlation)	254
14.3.3 余弦相似度	254
14.4 基于协同过滤的推荐算法	256

- 14.4.1 基于用户的协同过滤算法 256
- 14.4.2 基于项的协同过滤算法 258
- 14.5 利用协同过滤算法进行推荐 260
  - 14.5.1 导入用户-商品数据 260
  - 14.5.2 利用基于用户的协同过滤算法进行推荐 261
  - 14.5.3 利用基于项的协同过滤算法进行推荐 262
- 参考文献 264
- 15 基于矩阵分解的推荐算法 265
  - 15.1 矩阵分解 265
  - 15.2 基于矩阵分解的推荐算法 266
    - 15.2.1 损失函数 266
    - 15.2.2 损失函数的求解 266
    - 15.2.3 加入正则项的损失函数即求解方法 267
    - 15.2.4 预测 269
  - 15.3 利用矩阵分解进行推荐 270
    - 15.3.1 利用梯度下降对用户商品矩阵分解和预测 270
    - 15.3.2 最终的结果 272
  - 15.4 非负矩阵分解 273
    - 15.4.1 非负矩阵分解的形式化定义 274
    - 15.4.2 损失函数 274
    - 15.4.3 优化问题的求解 274
  - 15.5 利用非负矩阵分解进行推荐 277
    - 15.5.1 利用乘法规则进行分解和预测 277
    - 15.5.2 最终的结果 278
- 参考文献 279
- 16 基于图的推荐算法 280
  - 16.1 二部图与推荐算法 280
    - 16.1.1 二部图 280
    - 16.1.2 由用户商品矩阵到二部图 281
  - 16.2 PageRank算法 282
    - 16.2.1 PageRank算法的概念 282
    - 16.2.2 PageRank的两个假设 283
    - 16.2.3 PageRank的计算方法 283
  - 16.3 PersonalRank算法 285
    - 16.3.1 PersonalRank算法原理 285
    - 16.3.2 PersonalRank算法的流程 286
  - 16.4 利用PersonalRank算法进行推荐 288
    - 16.4.1 利用PersonalRank算法进行推荐 288
    - 16.4.2 最终的结果 291
- 参考文献 291
- 第五部分 深度学习
- 17 AutoEncoder 294
  - 17.1 多层神经网络 294
    - 17.1.1 三层神经网络模型 294
    - 17.1.2 由三层神经网络到多层神经网络 295
  - 17.2 AutoEncoder模型 296
    - 17.2.1 AutoEncoder模型结构 296
    - 17.2.2 AutoEncoder的损失函数 297
  - 17.3 降噪自编码器Denoising AutoEncoder 298
    - 17.3.1 Denoising AutoEncoder原理 298
    - 17.3.2 Denoising AutoEncoder实现 299
  - 17.4 利用Denoising AutoEncoders构建深度网络 302
    - 17.4.1 无监督的逐层训练 302
    - 17.4.2 有监督的微调 303

17.5 利用TensorFlow实现Stacked Denoising AutoEncoders	306
17.5.1 训练Stacked Denoising AutoEncoders模型	306
17.5.2 训练的过程	307
参考文献	308
18 卷积神经网络	309
18.1 传统神经网络模型存在的问题	309
18.2 卷积神经网络	311
18.2.1 卷积神经网络中的核心概念	311
18.2.2 卷积神经网络模型	312
18.3 卷积神经网络的求解	313
18.3.1 卷积层 (Convolution Layer)	313
18.3.2 下采样层 (Sub-Sampling Layer)	316
18.3.3 全连接层 (Fully-Connected Layer)	316
18.4 利用TensorFlow实现CNN	316
18.4.1 CNN的实现	316
18.4.2 训练CNN模型	320
18.4.3 训练的过程	321
参考文献	321
第六部分 项目实践	
19 微博精准推荐	324
19.1 精准推荐	324
19.1.1 精准推荐的项目背景	324
19.1.2 精准推荐的技术架构	325
19.1.3 离线数据挖掘	326
19.2 基于用户行为的挖掘	327
19.2.1 基于互动内容的兴趣挖掘	327
19.2.2 基于与博主互动的兴趣挖掘	328
19.3 基于相似用户的挖掘	329
19.3.1 基于“@”人的相似用户挖掘	329
19.3.2 基于社区的相似用户挖掘	329
19.3.3 基于协同过滤的相似用户挖掘	331
19.4 点击率预估	332
19.4.1 点击率预估的概念	332
19.4.2 点击率预估的方法	332
19.5 各种数据技术的效果	334
参考文献	335
附录A	336
附录B	341
• • • • •	(收起)

[Python机器学习算法\\_下载链接1](#)

标签

机器学习

python



计算机算法

计算机

编程

数学 · 计算机科学

Python

T-工业技术

## 评论

其实能抄出一本出版的书也算厉害……

-----  
代码有点old\_school啊，不太pythonic。部分算法细节还不错。

-----  
一堆公式的堆砌，各种重复，没有逻辑！

-----  
比同类国产书不知高到哪里去

-----  
强烈不推荐！觉得就是公式的堆砌，没有逻辑。看懂这些算法需要额外查找许多资料，看这本书完全没有意义。

-----  
Python的版本很重要吗？模型才重要啊

-----  
十分囫圇吞枣，过于浮于表面…

-----  
说不好是偏实战还是偏原理，感觉作者想均衡，但实际上都没说明白

-----  
有些章节确实是抄的《机器学习实战》，作者也把《机器学习实战》作为第一参考资料。还有一些章节是《机器学习实战》没有涉及的，例如 FM、Softmax 的实现，聚类算法等。就是感觉这本书的可以弄得大一些，代码看起来很不舒服。一些代码还可以写得再简洁一些，自己有的实现的部分只是为了帮助读者理解算法的思想，真正用的时候，应该用开源的、经过检验的代码，这部分最好要加上。不过也已经很不错了。

-----  
内容不错，代码编写的清楚，适合学习机器学习算法的学生使用

-----  
这本书对程序员很友好，每个算法都有对应的python实现，公式和代码对照分析，很快就能明白算法原理。  
不足的地方是算法推导的过程不够详细，翻看了很多资料才看懂推导逻辑。

-----  
[Python机器学习算法 下载链接1](#)

书评

-----  
[Python机器学习算法 下载链接1](#)