
揭开机器学习的面纱：最大熵模型100行代码实现[Python版]

*鑰*qingchuan 机器学习 2015-03-26 1,620 次浏览 GIS, MaxEnt, 最大熵 揭开机器学习的面纱：最大熵模型100行代码实现[Python版]已关闭评论

理论说明部分见上一篇：

最大熵模型简介[例子+推导+GIS求解]

为了是代码简短，方便阅读，去掉了很多健壮性检测的代码以及特殊处理。下面的代码实现的是：使用最基础GIS训练最大熵模型。GIS由于性能问题在实际中不适用，但是可以帮助我们理解最大熵训练到底在做什么。

```
#!/usr/bin/python
#coding=utf8
import sys;
import math;
from collections import defaultdict

class MaxEnt:
    def __init__(self):
        self._samples = []; #样本集, 元素是[y,x1,x2,...,xn]的元组
        self._Y = set([]); #标签集合,相当于去重之后的y
        self._numXY = defaultdict(int); #Key是(xi,yi)对，Value是count(xi,yi)
        self._N = 0;            #样本数量
        self._n = 0;            #特征对(xi,yi)总数量
        self._xyID = {};        #对(x,y)对做的顺序编号(ID), Key是(xi,yi)对,Value是ID
        self._C = 0;            #样本最大的特征数量,用于求参数时的迭代，见IIS原理说明
        self._ep_ = [];         #样本分布的特征期望值
        self._ep = [];          #模型分布的特征期望值
```

```
        self._w = [];           #对应n个特征的权值
        self._lastw = [];       #上一轮迭代的权值
        self._EPS = 0.01;       #判断是否收敛的阈值
    def load_data(self, filename):
        for line in open(filename, "r"):
            sample = line.strip().split("\t");
            if len(sample) < 2: #至少：标签+一个特征
                continue;
            y = sample[0];
            X = sample[1:];
            self._samples.append(sample); #labe + features
            self._Y.add(y); #label
            for x in set(X): #set给X去重
                self._numXY[(x, y)] += 1;
    def _initparams(self):
        self._N = len(self._samples);
        self._n = len(self._numXY);
        self._C = max([len(sample) - 1 for sample in self._samples]);
        self._w = [0.0] * self._n;
        self._lastw = self._w[:];
        self._sample_ep();
    def _convergence(self):
        for w, lw in zip(self._w, self._lastw):
            if math.fabs(w - lw) >= self._EPS:
                return False;
        return True;
    def _sample_ep(self):
        self._ep_ = [0.0] * self._n;
        #计算方法参见公式(20)
        for i, xy in enumerate(self._numXY):
            self._ep_[i] = self._numXY[xy] * 1.0 / self._N;
            self._xyID[xy] = i;
    def _zx(self, X):
        #calculate Z(X), 计算方法参见公式(15)
        ZX = 0.0;
        for y in self._Y:
            sum = 0.0;
            for x in X:
                if (x, y) in self._numXY:
                    sum += self._w[self._xyID[(x, y)]];
            ZX += math.exp(sum);
        return ZX;
    def _pyx(self, X):
        #calculate p(y|x), 计算方法参见公式(22)
        ZX = self._zx(X);
        results = [];
        for y in self._Y:
            sum = 0.0;
            for x in X:
                if (x, y) in self._numXY: #这个判断相当于指示函数的作用
                    sum +=      self._w[self._xyID[(x, y)]];
            pyx = 1.0 / ZX * math.exp(sum);
            results.append((y, pyx));
        return results;
    def _model_ep(self):
        self._ep = [0.0] * self._n;
        #参见公式(21)
        for sample in self._samples:
            X = sample[1:];
            pyx = self._pyx(X);
            for y, p in pyx:
```

```
                for x in X:
                    if (x, y) in self._numXY:
                        self._ep[self._xyID[(x, y)]] += p * 1.0 / self._N;
    def train(self, maxiter = 1000):
        self._initparams();
        for i in range(0, maxiter):
            print "Iter:%d..."%i;
            self._lastw = self._w[:]; #保存上一轮权值
            self._model_ep();
                #更新每个特征的权值
            for i, w in enumerate(self._w):
                #参考公式(19)
                self._w[i] += 1.0 / self._C * math.log(self._ep_[i] / self._ep[i]);
            print self._w;
            #检查是否收敛
            if self._convergence():
                break;
    def predict(self, input):
        X = input.strip().split("\t");
        prob = self._pyx(X)
        return prob;

if __name__ == "__main__":
    maxent = MaxEnt();
    maxent.load_data('data.txt');
    maxent.train();
    print maxent.predict("sunny\thot\thigh\tFALSE");
    print maxent.predict("overcast\thot\thigh\tFALSE");
    print maxent.predict("sunny\tcool\thigh\tTRUE");
    sys.exit(0);
```

训练数据来自各种天气情况下是否打球的例子：data.txt
其中字段依次是:

play outlook temperature humidity windy

部分运行结果：

```
71928268592, -1.2645221442480512, 1.3179713774210247, 1.39120228286492
Iter:273...
[2.1527896880880952, -2.6065160738693214, -3.0646845197820611, 2.75252
 -6.6375186004047677, -0.83336329140636078, -0.095284261881349711, 0.1
90802324, -1.2665758264458553, 1.3200420698923396, 1.3929989388592232,
Iter:274...
[2.155479887852108, -2.6100611147670869, -3.0692453017677539, 2.757057
28, -6.6476770277300776, -0.83485468076174207, -0.095093081980633112,
9040945894, -1.2686240834784444, 1.3221072470751107, 1.394790490242561
Iter:275...
[2.1581638857327206, -2.6135962244840023, -3.0737943138948682, 2.76157
134, -6.6578072850176433, -0.83634115573738355, -0.094903073890031331,
240021866144, -1.2706669460688358, 1.3241669404420342, 1.3965769690509
Iter:276...
[2.1608417150360353, -2.6171214645275134, -3.0783316227948547, 2.76606
552, -6.6679095400503776, -0.83782274916643651, -0.094714228500948031,
419872657004, -1.2727044446685232, 1.3262211811857085, 1.39835840700228
Iter:277...
[2.1635134087411996, -2.6206368958016415, -3.0828572944910917, 2.77055
569, -6.6779839590634511, -0.8392994935655933, -0.094526536786032053,
29028387938, -1.2747366094606956, 1.3282700002219789, 1.400134835500683
Iter:278...
[2.1661789995052967, -2.6241425786150172, -3.0873713944065613, 2.77502
67, -6.6880307067633789, -0.84077142113894132, -0.094339989798456023,
67920151711, -1.2767634703634096, 1.3303134281932312, 1.401906285640643
Iter:279...
[2.1688385196681441, -2.6276385726887788, -3.0918739873714016, 2.77947
231, -6.6980499463468153, -0.84223856378176043, -0.094154578671198766,
536975111465, -1.2787850570327159, 1.332351495471638, 1.403672788211208
Iter:280...
[2.1714920012569943, -2.631124937164341, -3.0963651376303387, 2.783913
09, -6.7080418395190602, -0.84370095308426651, -0.093970294616331065,
36616547121, -1.2808013988657383, 1.3343842321623569, 1.405434373700021
[('yes', 0.0041626518719793002), ('no', 0.99583734812802072)]
[('yes', 0.99436821023604471), ('no', 0.0056317897639553702)]
[('yes', 1.4464465173635744e-07), ('no', 0.99999985535534819)]
[root@iZ25ttq9nvpZ maxent-python]#
```

本文由《纯净的天空》出品，原文地址:https://vimsky.com/article/776.html，未经允许，请勿转载。

- 最大熵模型简介[例子+推导+GIS求解]

上一篇：协同过滤之ALS-WR算法
下一篇：常用机器学习算法的点睛之笔

相关文章

最大熵模型简介[例子+推导+GIS求解]

揭开机器学习的面纱：SVM 100行代码实现[Python版]


pyspark RandomForest的分类和回归示例


pyspark GBDT分类和回归示例


Spark中ml和mllib的区别


pyspark LogisticRegressionModel用法示例


pyspark LDA模型示例


Spark机器学习库指南[Spark 1.3.1版]——聚类(Clustering)


Spark机器学习库指南[Spark 1.3.1版]——数据类型(Data Types)


Spark机器学习库指南[Spark 1.3.1版]——特征提取和转换(Feature extraction and transformation)

- 热门文章
- 最新发布
- 随机推荐

- Wordpress自动批量设置特色图像(Featured Image)
- Wordpress中解决Markdown和Latex冲突问题
- Linux Shell命令之文本内容分析
- Spark机器学习库指南[Spark 1.3.1版]——协同过滤(Collaborative Filtering)
- Spark的Cache和Checkpoint
- 为什么L1稀疏，L2平滑？
- 那些年踩过的坑。。。
- Spark二进制文件读写
- SDCC2015 机器学习在美团用户画像中的应用PPT
- Scala编程常见问题整理【十一】
- Wordpress通过url批量自动插入文章
- Apache Spark和Apache Storm的区别
- Stackoverflow热门技术排行榜Top180
- Java编程常见问题集锦【三】

- Java编程常见问题集锦【四】

- MySql（MariaDB）在小内存VPS上崩溃，该怎么办？
- 在低内存虚拟机上启动MySQL
- 深度学习库Keras入门
- Spark任务提交(Spark Submit)
- Spark Streaming入门
- pyspark卡方特征选择ChiSqSelector用法示例
- 数据库事务Spring @Transactional注解失效原因分析
- Java BigDecimal浮点数运算--如何保证运算精度不溢出
- Spark中ml和mllib的区别
- pyspark LDA模型示例
- pyspark GBDT分类和回归示例
- XGBoost调参注意事项
- Mac系统安装XGboost
- pyspark RandomForest的分类和回归示例
- Googlebot大量请求wp-login.php?redirect_to=解决办法

- 揭开机器学习的神秘面纱：一张图弄懂协同过滤
- Spark二进制文件读写
- 揭开机器学习的面纱：最大熵模型100行代码实现[Python版]
- 深度优先搜索算法之非递归实现及实例源码(原创)
- Swift简介
- 基于哈夫曼(haffuman)算法的文件压缩的实现（C语言）（原创）
- 一张图记住vim常用命令
- 中缀表达式转化为后缀表达式[附C++源码]（原创）
- 揭开机器学习的面纱：SVM 100行代码实现[Python版]
- Wordpress数据库错误:Lost connection to MySQL server
- Scala编程常见问题整理【十五】
- C++ undefined reference/unresolved external symbol问题原因分析
- 可以求最小值的栈的实现（C语言）（原创）
- 机器学习开源库[转]
- 深度学习库Keras入门