

刘建平Pinard

十年码农，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

用hmmlearn学习隐马尔科夫模型HMM

在之前的HMM系列中，我们对隐马尔科夫模型HMM的原理以及三个问题的求解方法做了总结。本文我们就从实践的角度用Python的hmmlearn库来学习HMM的使用。关于hmmlearn的更多资料在[官方文档](#)有介绍。

1. hmmlearn概述

hmmlearn安装很简单，"pip install hmmlearn"即可完成。

hmmlearn实现了三种HMM模型类，按照观测状态是连续状态还是离散状态，可以分为两类。GaussianHMM和GMMHMM是连续观测状态的HMM模型，而MultinomialHMM是离散观测状态的模型，也是我们在HMM原理系列篇里面使用的模型。

对于MultinomialHMM的模型，使用比较简单，"startprob\_"参数对应我们的隐藏状态初始分布 $\Pi$ ，"transmat\_"对应我们的状态转移矩阵 $A$ ，"emissionprob\_"对应我们的观测状态概率矩阵 $B$ 。

对于连续观测状态的HMM模型，GaussianHMM类假设观测状态符合高斯分布，而GMMHMM类则假设观测状态符合混合高斯分布。一般情况下我们使用GaussianHMM即高斯分布的观测状态即可。以下对于连续观测状态的HMM模型，我们只讨论GaussianHMM类。


在GaussianHMM类中，"startprob\_"参数对应我们的隐藏状态初始分布 $\Pi$ ，"transmat\_"对应我们的状态转移矩阵 $A$ ，比较特殊的是观测状态概率的表示方法，此时由于观测状态是连续值，我们无法像MultinomialHMM一样直接给出矩阵 $B$ ，而是采用给出各个隐藏状态对应的观测状态高斯分布的概率密度函数的参数。

如果观测序列是一维的，则观测状态的概率密度函数是一维的普通高斯分布。如果观测序列是 $N$ 维的，则隐藏状态对应的观测状态的概率密度函数是 $N$ 维高斯分布。高斯分布的概率密度函数参数可以用 $\mu$ 表示高斯分布的期望向量， $\Sigma$ 表示高斯分布的协方差矩阵。在GaussianHMM类中，"means"用来表示各个隐藏状态对应的高斯分布期望向量 $\mu$ 形成的矩阵，而"covars"用来表示各个隐藏状态对应的高斯分布协方差矩阵 $\Sigma$ 形成的三维张量。

2. MultinomialHMM实例

下面我们使用我们在HMM系列原理篇中的例子来使用MultinomialHMM跑一遍。

首先建立HMM的模型：



```
import numpy as np
from hmmlearn import hmm

states = ["box 1", "box 2", "box3"]
n_states = len(states)

observations = ["red", "white"]
n_observations = len(observations)

start_probability = np.array([0.2, 0.4, 0.4])

transition_probability = np.array([
    [0.5, 0.2, 0.3],
    [0.3, 0.5, 0.2],
    [0.2, 0.3, 0.5]
])

emission_probability = np.array([
    [0.5, 0.5],
    [0.4, 0.6],
    [0.7, 0.3]
])

model = hmm.MultinomialHMM(n_components=n_states)
```

公告

★珠江追梦，饮岭南茶，恋鄂北家★  
昵称：刘建平Pinard  
园龄：1年5个月  
粉丝：1057  
关注：13  
+加关注

2018年3月						
<	日	一	二	三	四	五
	25	26	27	28	1	2
	4	5	6	7	8	9
	11	12	13	14	15	16
	18	19	20	21	22	23
	25	26	27	28	29	30
	1	2	3	4	5	6

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

```
model.startprob_ = start_probability
model.transmat_ = transition_probability
model.emissionprob_ = emission_probability
```



现在我们来跑一跑HMM问题三维特比算法的解码过程，使用和原理篇一样的观测序列来解码，代码如下：

```
seen = np.array([[0,1,0]]).T
logprob, box = model.decode(seen, algorithm="viterbi")
print("The ball picked:", " ", ".join(map(lambda x: observations[x], seen)))
print("The hidden box", " ", ".join(map(lambda x: states[x], box)))
```

输出结果如下：

```
('The ball picked:', 'red, white, red')
('The hidden box', 'box3, box3, box3')
```

可以看出，结果和我们原理篇中的手动计算的结果是一样的。

也可以使用predict函数，结果也是一样的，代码如下：

```
box2 = model.predict(seen)
print("The ball picked:", " ", ".join(map(lambda x: observations[x], seen)))
print("The hidden box", " ", ".join(map(lambda x: states[x], box2)))
```

大家可以跑一下，看看结果是否和decode函数相同。

现在我们再来看看求HMM问题一的观测序列的概率的问题，代码如下：

```
print model.score(seen)
```

输出结果是：

```
-2.03854530992
```

要注意的是score函数返回的是以自然对数为底的对数概率值，我们在HMM问题一中手动计算的结果是未取对数的原始概率是0.13022。对比一下：

$$\ln 0.13022 \approx -2.0385$$

现在我们再看看HMM问题二，求解模型参数的问题。由于鲍姆-韦尔奇算法是基于EM算法的近似算法，所以我们需要多跑几次，比如下面我们跑三次，选择一个比较优的模型参数，代码如下：

```
import numpy as np
from hmmlearn import hmm

states = ["box 1", "box 2", "box3"]
n_states = len(states)

observations = ["red", "white"]
n_observations = len(observations)
model2 = hmm.MultinomialHMM(n_components=n_states, n_iter=20, tol=0.01)
X2 = np.array([[0,1,0,1],[0,0,0,1],[1,0,1,1]])
model2.fit(X2)
print model2.startprob_
print model2.transmat_
print model2.emissionprob_
print model2.score(X2)
model2.fit(X2)
print model2.startprob_
print model2.transmat_
print model2.emissionprob_
print model2.score(X2)
model2.fit(X2)
print model2.startprob_
print model2.transmat_
print model2.emissionprob_
print model2.score(X2)
```



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 - 298466

排名 - 614

阅读排行榜

- 1. 梯度下降 ( Gradient Descent ) 小结(94544)
- 2. 梯度提升树(GBDT)原理小结(45020)
- 3. 线性判别分析LDA原理总结(30541)
- 4. scikit-learn决策树算法类库使用小结(27497)
- 5. 谱聚类 ( spectral clustering ) 原理总结(20496)

评论排行榜

- 1. 梯度提升树(GBDT)原理小结(79)
- 2. 谱聚类 ( spectral clustering ) 原理总结(62)
- 3. 梯度下降 ( Gradient Descent ) 小结(60)
- 4. 卷积神经网络(CNN)反向传播算法(56)
- 5. 集成学习之Adaboost算法原理小结(50)

推荐排行榜

- 1. 梯度下降 ( Gradient Descent ) 小结(41)
- 2. 集成学习原理小结(14)
- 3. 卷积神经网络(CNN)反向传播算法(14)
- 4. 集成学习之Adaboost算法原理小结(13)
- 5. 协同过滤推荐算法总结(11)

结果这里就略去了，最终我们会选择分数最高的模型参数。

以上就是用MultinomialHMM解决HMM模型三个问题的方法。

### 3. GaussianHMM实例

下面我们再给一个GaussianHMM的实例，这个实例中，我们的观测状态是二维的，而隐藏状态有4个。因此我们的“means”参数是 $4 \times 2$ 的矩阵，而“covars”参数是 $4 \times 2 \times 2$ 的张量。

建立模型如下：

```
startprob = np.array([0.6, 0.3, 0.1, 0.0])
# The transition matrix, note that there are no transitions possible
# between component 1 and 3
transmat = np.array([[0.7, 0.2, 0.0, 0.1],
                     [0.3, 0.5, 0.2, 0.0],
                     [0.0, 0.3, 0.5, 0.2],
                     [0.2, 0.0, 0.2, 0.6]])

# The means of each component
means = np.array([[0.0, 0.0],
                  [0.0, 11.0],
                  [9.0, 10.0],
                  [11.0, -1.0]])

# The covariance of each component
covars = .5 * np.tile(np.identity(2), (4, 1, 1))

# Build an HMM instance and set parameters
model3 = hmm.GaussianHMM(n_components=4, covariance_type="full")

# Instead of fitting it from the data, we directly set the estimated
# parameters, the means and covariance of the components
model3.startprob_ = startprob
model3.transmat_ = transmat
model3.means_ = means
model3.covars_ = covars
```

注意上面有个参数covariance\_type，取值为“full”意味所有的 $\mu, \Sigma$ 都需要指定。取值为“spherical”则 $\Sigma$ 的非对角线元素为0，对角线元素相同。取值为“diag”则 $\Sigma$ 的非对角线元素为0，对角线元素可以不同，“tied”指所有的隐藏状态对应的观测状态分布使用相同的协方差矩阵 $\Sigma$

我们现在跑一跑HMM问题—解码的过程，由于观测状态是二维的，我们用的三维观测序列，所以这里的输入是一个 $3 \times 2 \times 2$ 的张量，代码如下：

```
seen = np.array([[1.1, 2.0], [-1, 2.0], [3, 7]])
logprob, state = model.decode(seen, algorithm="viterbi")
print state
```

输出结果如下：

[0 0 1]

再看看HMM问题—对数概率的计算：

```
print model3.score(seen)
```

输出如下：

-41.1211281377

以上就是用hmmlearn学习HMM的过程。希望可以帮到大家。

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


分类: 0083. 自然语言处理

好文要顶

关注我

收藏该文





刘建平Pinard

关注 - 13

粉丝 - 1057

+加关注

« 上一篇：[隐马尔科夫模型HMM（四）维特比算法解码隐藏状态序列](#)

» 下一篇：[条件随机场CRF\(一\)从随机场到线性链条件随机场](#)

posted @ 2017-06-13 16:24 刘建平Pinard 阅读(4852) 评论(2) 编辑 收藏

评论列表

- #1楼 2017-11-23 02:26 ZuoZuoHao

好赞，期待继续更新

支持(0) 反对(0)
- #2楼 2018-01-27 18:34 dahu1

正学着hmm呢，就看到此篇好文，感谢

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

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！
- 【缅怀】传奇谢幕，回顾霍金76载传奇人生
- 【推荐】腾讯云校园拼团福利，1核2G服务器10元/月！
- 【活动】2050 科技公益大会 - 年青人因科技而团聚

 腾讯云

0基础建站 网站模板9元起

五合一建站套餐 满足多场景需求

立即抢购

- 最新IT新闻：
- Linux基金会宣布开放物联网ACRN管理程序

· 黑莓签约成为微软移动APP“保镖” 股价大涨

· Google Pay现在可以处理城市交通票支付

· 低价+社交 拼多多“釜底抽薪”阿里京东

· 解密Uber自动驾驶系统：多重传感器保护为何撞死人？
- » 更多新闻...

 阿里云

新购满返 ¥6000 封顶

广告

- 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...