

针对序列数据创建隐马尔科夫模型

- import datetime
- import numpy as np
- import matplotlib.pyplot as plt
- from hmmlearn.hmm import GaussianHMM
- from Chapter08.convert_to_timeseries import convert_data_to_timeseries
- ''' 功能作用：针对序列数据创建隐马尔科夫模型
- 希望预测明天的天气是晴天、阴天或下雨。为了实现预测，需要查看所有的参数，例如温度、气压等，而潜在的状态是隐藏的。这里，潜在的状态是指3个可选状态：晴天、阴天或下雨
- 步骤：
 - 1.利用np.loadtxt导入数据文件
 - 2.分割并且提取文件中的变量
 - 3.开始创建HMM模型,选择适合的参数
 - 4.预测被隐藏的状态,并且输入隐藏的状态信息'''
- # 输入数据文件
- input_file = 'data_hmm.txt' data = np.loadtxt(input_file, delimiter=',')
- # column_stack：代表 矩阵中添加列：
- # 排列训练数据 ,取第二列的数据
- X = np.column_stack([data[:,2]])
- '''用4个成分创建并训练HMM。成分的个数是一个需要进行选择的超参数。这里选择4个成分，也就意味着用4个潜在状态生成数据'''
- # 创建并训练 高斯HMM模型
- print ("\nTraining HMM....")
- num_components = 4
- model = GaussianHMM(n_components=num_components, covariance_type="diag", n_iter=1000)
- model.fit(X)
- # 预测HMM 的隐藏状态 计算这些隐藏状态的均值和方差
- hidden_states = model.predict(X)
- print ("\nMeans and variances of hidden states:")
- for i in range(model.n_components):
 - print ("\nHidden state", i+1)
 - print ("Mean =", round(model.means_[i][0], 3))
 - print ("Variance =", round(np.diag(model.covars_[i])[0], 3))
- # 用模型生成数据
- num_samples = 1000
- samples, _ = model.sample(num_samples)
- plt.plot(np.arange(num_samples), samples[:,0], c='black')
- plt.title('Number of components = ' + str(num_components))

- plt.show()

幕布 - 思维概要整理工具
