使用HMM进行中文分词

序列标签

首先我们先给序列数据打上标签。BMES，B代表词开始，M代表词的中间，E代表词的结尾。S代表单字词。比如下面：

```
致B以E诚B挚E的S问B候E和S良B好E的S祝B愿E！S
```

在本次实战中，我们使用98人民日报标注语料进行训练。

构建HMM

通过以上的序列标注，那么我们可以得到这个HMM模型：

- **状态空间**为{B,E,M,S}
- 每个字就是模型中的观测，所以**观测空间**为语料中的所有中文字

两个空间完了，还需要三个矩阵。

- 状态转移概率矩阵：这个需要统计模型中所有状态的转换进而计算概率即可。
- 输出观测概率矩阵：这个需要统计四种状态下出来各个观测（也就是各个字）的概率。
- 初始状态概率：模型在初始时，各状态出现的概率。

至此我们的一个HMM模型就构建完成了。

实际分词

我们通过HMM的预测问题，输入进去一串中文字符串，然后得到一串对应的标注的序列，最终根据标注的序列，进行分词。BME是一个词，S是一个单字词，这样就实现了一个简单版本的中文分词。

代码实现

1. 一些参数的设置

```
# 序列化文件夹
```

```
model_path = 'model/hmm.model'
default_probability = 0.000000001
# 转移概率矩阵
trans_mat = {}
# 观测概率矩阵
emit_mat = {}
# 初始概率矩阵
init_vec = {}
# 状态集合
state_set = set()
# 观测集合
observation_set = set()
data_path = 'data/199801人民日报.data'
```

2. train

这一步实际上就是进行一些计算和统计，将HMM所需要的一些空间、概率矩阵等等计算出来。具体代码可参考提供的源代码train方法。

3. predict

这一部分是HMM的预测问题，使用的是Viterbi算法求解出需要进行分词的字符串对应的标注。具体的原理可参考视频中关于维特比算法的部分，具体的代码请参考提供的源代码中的predict方法。