CH09 EM算法及其推广

前言

章节目录

导读

- **概率模型**有时既含有观测变量，又含有隐变量或潜在变量。这句很重要，有时候我们能拿到最后的观测结果，却拿不到中间的过程记录。
- 这章如果看三硬币有疑问，可以往后继续看，看到高斯混合模型，然后再回头理解三硬币。有不理解的地方，可以重新看对应问题的定义，重新理解各个符号的意义，因为这章开始，需要分析的问题和之前的分类问题有差异，任务不同了要理解需求。希望对学习有帮助。
- EM算法可以用于**生成模型**的非监督学习，EM算法是个一般方法，不具有具体模型。

  > EM算法是一种迭代算法，用于含有隐变量的概率模型的极大似然估计，或极大后验概率估计。

  本书CH12在对比各种模型的策略的时候，从这章开始，学习策略都是MLE，损失函数都是对数似然损失。体现了这一类问题的共性与联系。
- 这里面注意体会不同变量的大小以及对应的取值范围。
- 一个$m \times n \times k$的矩阵可能可以划分成n个$m \times k$的形式，这点理解下。
- 涉及混合模型的部分推导有很多求和，注意体会是按照**样本**做的，还是按照**模型**做的，也就是操作的域。
- 如果对PDF，高斯分布，边缘概率分布，协方差矩阵不清楚，可以在这个章节从GMM的角度扩展阅读下，一定会有收获。
- 似然和概率的关系可以推广了解，这章关于概率和似然的符号表示，可能会有点看不懂，比如P_{157}（第二版P_{177}）中的部分表述。可以参考引用内容[1]，概率和似然是同样的形式描述的都是**可能性**，$P(Y|\theta)$是一个两变量的函数，**似然**是给定结果，求参数可能性；**概率**是给定参数求结果可能性。
  是不是可以认为，似然和概率分别对应了Train和Predict的过程？感受下。

  > Suppose you have a probability model with parameters θ.
  > $p(x|\theta)$ has two names.
  > It can be called the **probability of** x (given θ),
  > or the **likelihood of** θ (given that x was observed).

  书中对应的描述是

  > 假设给定观测数据Y，其概率分布是。。。那么不完全数据Y的似然函数是。。。对数似然函数。。。。就这段，后面符号说明部分有引用。

- 学习过程中注意**观测数据**在EM算法每次迭代中的意义。
- GMM中注意区分α_k和γ_{jk}的差异，直觉上都有一种归属的感觉，γ_{jk}是二值函数，α_k是一种概率的表示。γ_{jk}是one-hot encoding(also: 1-of-K representation)，还有$\hat{\gamma}_{jk}$这个是个估计注意和γ_{jk}的关系。
- GMM这里面实际上还涉及到一个概念叫做凸组合(Convex Combination)[2]，是凸几何领域的一个概念，点的线性组合，所有系数都非负且和为1。点集的凸包等价于该点集的凸组合。
- 无论是三硬币还是GMM，采样的过程都是如下:

  > 1. Sample $z_i \sim p(z|\pi)$
  > 2. Sample $x_i \sim p(x|\pi)$

  注意，这里用到了π，在强化学习中，随机性策略$\pi(x,a)$表示为状态x下选择动作a的概率。
- 关于EM算法的解释
  注意这里EM不是模型，是个一般方法，不具有具体的模型，这点前面已经提到
  1. PRML
     $kmeans \rightarrow GMM \rightarrow EM$
     所以，EM应用举例子为kmeans也OK。而且，西瓜书P_{165}上有说，`k均值聚类算法就是一个典型的EM算法`
  2. 统计学习方法
     1. $MLE \rightarrow B$
     2. F函数的极大-极大算法
- 这个repo里面实现了BMM算法和GMM算法两种混合模型。
- HMM也是Discrete **Dynamic Model**，从图模型角度考虑，可以发现HMM和卡尔曼滤波以及粒子滤波深层之间的联系。这部分内容在PRML中有讨论。
- 书中图9.1说一下，可以参考CH08的部分内容，关于Bregman distance的那部分说明。
- HMM作了两个基本假设，实际上是在说在图模型中，存在哪些**边**。
- 第二版里面增加了聚类方法的描述

符号说明

> 一般地，用Y表示观测随机变量的数据，Z表示隐随机变量的数据。Y和Z一起称为**完全数据**(complete-data)，观测数据Y又称为**不完全数据**(incomplete-data)

上面这个概念很重要，Dempster在1977年提出EM算法的时候文章题目就是《Maximum likelihood from incomplete data via the EM algorithm》，具体看书中本章参考文献[1]

> 假设给定观测数据Y，其概率分布是$P(Y|\theta)$，其中θ是需要估计的模型参数
> 那么不完全数据Y的似然函数是$P(Y|\theta)$，对数似然函数是$L(\theta)=\log P(Y|\theta)$
>
> 假设Y和Z的联合概率分布是$P(Y,Z|\theta)$，那么完全数据的对数似然函数是$\log P(Y,Z|\theta)$

上面这部分简单对应一下，这里说明一下，你看到下面概率分布和似然函数形式看起来一样。在概率中，θ已知，求Y，在似然函数中通过已知的Y去求θ

	观测数据Y	不完全数据Y	
不完全数据Y	概率分布$P(Y\|\theta)$	似然函数$P(Y\|\theta)$	对数似然函数$\log P(Y\|\theta)$
完全数据 (Y,Z)	Y和Z的联合概率分布 $P(Y,Z\|\theta)$	似然函数 $P(Y,Z\|\theta)$	对数似然函数 $\log P(Y,Z\|\theta)$

观测数据Y

有一点要注意下, 这里没有出现X, 在**9.1.3**节中有提到一种理解

> - 有时训练数据只有输入没有对应的输出$(x_1,\cdot),(x_2,\cdot),\dots,(x_N,\cdot)$，从这样的数据学习模型称为非监督学习问题。
> - EM算法可以用于生成模型的非监督学习。
> - 生成模型由联合概率分布$P(X,Y)$表示，可以认为非监督学习训练数据是联合概率分布产生的数据。X为观测数据，Y为未观测数据。

有时候，只观测显变量看不到关系，就需要把隐变量引进来。

混合模型

书中用三硬币模型做为引子，在学习这部分内容的时候，注意体会观测数据的作用。

伯努利混合模型(三硬币模型)

问题描述

书中用例子来介绍EM算法的问题，并给出了EM算法迭代求解的过程，具体例子描述见**例9.1**，这块如果不懂，可以跳过，看完后面高斯混合模型再回来看。

问题的描述过程中有这样一句：独立的重复n次实验(这里$n=10$)，观测结果如下:

`1,1,0,1,0,0,1,0,1,1`

思考上面这个观测和`1,1,1,1,1,1,0,0,0,0`有区别么?

没有任何信息的前提下，我们得到上面的观测数据可以假定是一个**二项分布**的形式，参数$n=10,p=0.6$

把$k=6$次成功分布在$n=10$次试验中有$C(10,6)$种可能.

所以上面两个观测序列，可能出自同一个模型。在这个问题的求解上是没有区别的，测试案例$test_t91$做了这个说明，可以参考。

我们通过一段代码来生成这个数据

```
import numpy as np

p = 0.6
n = 10
# np.random.seed(2018)
flag_a = 1
flag_b = 1
cnt = 0
while flag_a or flag_b:
    tmp = np.random.binomial(1, p, n)
    if (tmp == np.array([1,1,1,1,1,1,0,0,0,0])).all():
        flag_a = 0
        print("[1,1,1,1,1,1,0,0,0,0] at %d\n" % cnt)
    if (tmp == np.array([1,1,0,1,0,0,1,0,1,1])).all():
        flag_b = 0
        print("[1,1,0,1,0,0,1,0,1,1] at %d\n" % cnt)
    cnt += 1
```

实际上题目的描述中说明了观测数据生成的过程，这些参数是未知的，所以需要对这些参数进行估计。

解的过程记录在这里。

三硬币模型可以写作

$$\begin{aligned}P(y|\theta)&=\sum_z P(y,z|\theta)\\&=\sum_z P(z|\theta)P(y|z,\theta)\\&=\pi p^y(1-p)^{1-y}+(1-\pi)q^y(1-q)^{1-y}\end{aligned}$$

以上

1. 随机变量y是观测变量，表示一次试验观测的结果是**1或0**
2. 随机变量z是隐变量，表示未观测到的掷硬币A的结果
3. $\theta=(\pi,p,q)$是模型参数
4. 这个模型是**以上数据**(1,1,0,1,0,0,1,0,1,1)的生成模型

观测数据表示为$Y=(Y_1,Y_2,Y_3,\dots,Y_n)^T$, 未观测数据表示为$Z=(Z_1,Z_2,Z_3,\dots,Z_n)^T$, 则观测数据的似然函数为

> 其实觉得这里应该是小写的$y=(y_1,y_2,\dots,y_n),z=(z_1,z_2,\dots,z_n)$

$$P(Y|\theta)=\sum_Z P(Z|\theta)P(Y|Z,\theta)$$

注意这里的求和是下面的"+"描述的部分

即

$$P(Y|\theta)=\prod_{j=1}^{n}[\pi p^{y_j}(1-p)^{1-y_j}+(1-\pi)q^{y_j}(1-q)^{1-y_j}]$$

注意这里连乘是$Y\rightarrow y_j$出来的, 不理解看似然定义.

考虑求模型参数$\theta=(\pi,p,q)$的极大似然估计, 即

$$\hat{\theta}=\arg\max_{\theta}\log P(Y|\theta)$$

这个题目的标准答案实际上也是未知的，因为可能生成这样的观测的假设空间太大。

三硬币模型的EM算法

1.初值

EM算法首选参数初值，记作$\theta^{(0)}=(\pi^{(0)},p^{(0)},q^{(0)})$, 然后迭代计算参数的估计值。

如果第i次迭代的模型参数估计值为$\theta^{(i)}=(\pi^{(i)},p^{(i)},q^{(i)})$

2.E步

那么第$i+1$次迭代的模型参数估计值表示为

$$\mu_j^{i+1}=\frac{\pi^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j}}{\pi^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j}+(1-\pi^{(i)})(q^{(i)})^{y_j}(1-q^{(i)})^{1-y_j}}$$

因为是硬币，只有0，1两种可能，所有有上面的表达。

这个表达方式还可以拆成如下形式

$$\mu_j^{i+1}=\begin{cases}\frac{\pi^{(i)}p^{(i)}}{\pi^{(i)}p^{(i)}+(1-\pi^{(i)})q^{(i)}} & ,y_j=1\\ \frac{\pi^{(i)}(1-p^{(i)})}{\pi^{(i)}(1-p^{(i)})+(1-\pi^{(i)})(1-q^{(i)})} & ,y_j=0\end{cases}$$

所以, 这步(求μ_j)干了什么，样本起到了什么作用?

这一步，通过假设的参数，计算了不同的样本对假设模型的响应(μ_j)，注意这里因为样本(y_j)是二值的，所以，用$\{y_j,1-y_j\}$构成了one-hot的编码，用来表示样本归属的假设。

以上，有点绕。

这一步是什么的期望？书中有写，**观测数据来自硬币B的概率, 在二项分布的情况下, 响应度和概率是一个概念.** 这个说明，有助于后面M步公式的理解。

3.M步

$$
\begin{aligned}
\pi^{(i+1)} &= \frac{1}{n}\sum_{j=1}^{n}\mu_j^{(i+1)} \\
p^{(i+1)} &= \frac{\sum_{j=1}^{n}\mu_j^{(i+1)}y_j}{\sum_{j=1}^{n}\mu_j^{(i+1)}} \\
q^{(i+1)} &= \frac{\sum_{j=1}^{n}(1-\mu_j^{(i+1)})y_j}{\sum_{j=1}^{n}(1-\mu_j^{(i+1)})}
\end{aligned}
$$

上面，红色部分的公式从`观测数据是来自硬币B的概率`这句来理解。

初值影响

这个例子里面0.5是个合理又牛逼的初值。迭代收敛的最后结果是(0.5, 0.6, 0.6)

这个结果说明，如果A是均匀的，那么一个合理的解就是B，C是同质的。他们的分布情况和观测的分布一致。

在测试案例$test_e91$中有计算这部分的结果，注意看，这种简单的模型其实收敛的很快。

p,q 含义

这里面p对应了$A=1$，$B=1$，q对应了$A=0$，$C=1$

这三个公式可以改写成如下形式:

$$
\begin{aligned}
\pi^{(i+1)} &= \frac{1}{n}\sum_{j=1}^{n}\mu_j^{(i+1)} \\
p^{(i+1)} &= \frac{\sum_{j=1}^{n}\mu_j^{(i+1)}y_j}{\sum_{j=1}^{n}(\mu_j^{(i+1)}y_j+\mu_j^{(i+1)}(1-y_j))} \\
q^{(i+1)} &= \frac{\sum_{j=1}^{n}(1-\mu_j^{(i+1)})y_j}{\sum_{j=1}^{n}((1-\mu_j^{(i+1)})y_j+(1-\mu_j^{(i+1)})(1-y_j))}
\end{aligned}
$$

π的表达式回答这样一个问题: 刷了这么多样本，拿到一堆数，那么π应该是多少，均值是个比较好的选择。

p的表达式回答这样一个问题: 如果我知道每个结果y_j以μ_j的可能来自硬币B(A=1)，那么用这些数据刷出来他可能是正面的概率。这里面μ_j对应了$A=1$

q的表达式同理，其中$1-\mu_j$对应了$A=0$

到后面讲高斯混合模型的时候，可以重新审视这里

$$
\begin{aligned}
&\alpha_0 \leftrightarrow \pi \\
&\mu_0 \leftrightarrow p^{y_j}(1-p)^{1-y_j} \\
&\alpha_1 \leftrightarrow 1-\pi \\
&\mu_1 \leftrightarrow q^{y_j}(1-q)^{1-y_j}
\end{aligned}
$$

以上对应了包含两个分量的伯努利混合模型, BMM, 包含四个参数, 因为α_k满足等式约束, 所以通常会有三个参数, 另外参见习题9.3中有提到`两个分量的高斯混合模型的五个参数`实际上也是因为等式约束.

bmm.py对伯努利混合模型做了实现, 有几点说明一下:

1. $(p^{(i)})^{y_i}(1-p^{(i)})^{1-y_i}$ 这个表达式对应了伯努利分布的概率密度，可以表示成矩阵乘法，尽量不要用for，效率会差
2. 书中e_{91}的表达中，采用了π, p, q来表示，注意在题目的说明部分有说明三个符号的含义
3. 实际上不怎么抛硬币，但是0-1的伯努利分布很多，在书中算法9.4部分，有这样一个说明:

> 当参数θ的维数为$d(d \geq 2)$的时候，可以采用一种特殊的GEM算法，它将算法的M步分解成d次条件极大化，每次只改变参数向量的一个分量，其余量不改变。

EM算法另外视角

> 输入: 观测变量数据Y，隐变量数据Z，联合分布$P(Y,Z|\theta)$，条件分布$P(Z|Y,\theta)$
>
> 输出: 模型参数θ
>
> 1. 选择参数的初值$\theta^{(0)}$，开始迭代
> 2. E步：记$\theta^{(i)}$为第 i 次迭代参数θ的估计值，在第$i+1$次迭代的E步，计算
>
> $$\begin{aligned} Q(\theta,\theta^{(i)}) =& E_Z[\log P(Y,Z|\theta)|Y,\theta^{(i)}] \\ =& \sum_Z \log P(Y,Z|\theta) P(Z|Y,\theta^{(i)}) \end{aligned}$$
>
> 3. M步
>
> 求使$Q(\theta,\theta^{(i)})$最大化的θ，确定第$i+1$次迭代的参数估计值
>
> $$\theta^{(i+1)} = \arg\max_\theta Q(\theta,\theta^{(i)})$$

Q 函数

注意Q函数的定义，可以帮助理解上面E步中的求和表达式

完全数据的**对数似然函数**$\log P(Y,Z|\theta)$**关于**给定观测数据Y的当前参数$\theta^{(i)}$下对为观测数据Z的**条件概率分布**$P(Z|Y,\theta^{(i)})$**的期望**称为Q函数。

BMM的EM算法

> 输入: 观测变量数据$y_1,y_2,\dots,y_N$，伯努利混合模型
>
> 输出: 伯努利混合模型参数
>
> 1. 选择参数的初始值开始迭代，$2K$ 个参数
> 2. E步:
>
> $$\hat{\gamma}_{jk} = \frac{\alpha_k Bern(y_j|\theta_k)}{\sum_{k=1}^K \alpha_k Bern(y_j|\theta_k)} = \frac{\alpha_k \mu_k^{y_j}(1-\mu_k)^{1-y_j}}{\sum_{k=1}^K \alpha_k \mu_k^{y_j}(1-\mu_k)^{1-y_j}}, j=1,2,\dots,N; k=1,2,\dots,K$$
>
> 3. M步:
>
> $$\hat{\mu}_k = \frac{\sum_{j=1}^N \hat{\gamma}_{jk} y_j}{\sum_{j=1}^N \hat{\gamma}_{jk}}$$
>
> $$\hat{\alpha}_k = \frac{n_k}{N}$$

目标函数L

$$L(\theta) = \log P(Y|\theta) = \log \sum_Z P(Y,Z|\theta) = \log(\sum_Z P(Y|Z,\theta)P(Z|\theta))$$

目标函数是不完全数据的对数似然

EM算法导出

书中这部分内容回答为什么EM算法能近似实现对观测数据的极大似然估计?

$$\begin{aligned} L(\theta) - L(\theta^{(i)}) &= \log\left(\sum_Z P(Z|Y,\theta^{(i)}) \frac{P(Y|Z,\theta)P(Z|\theta)}{P(Z|Y,\theta^{(i)})}\right) - \log P(Y|\theta^{(i)}) \\ &\geq \sum_Z P(Z|Y,\theta^{(i)}) \log \frac{P(Y|Z,\theta)P(Z|\theta)}{P(Z|Y,\theta^{(i)})} - \log P(Y|\theta^{(i)}) \\ &= \sum_Z P(Z|Y,\theta^{(i)}) \log \frac{P(Y|Z,\theta)P(Z|\theta)}{P(Z|Y,\theta^{(i)})} - \sum_Z P(Z|Y,\theta^{(i)}) \log P(Y|\theta^{(i)}) \\ &= \sum_Z P(Z|Y,\theta^{(i)}) \log \frac{P(Y|Z,\theta)P(Z|\theta)}{P(Z|Y,\theta^{(i)})P(Y|\theta^{(i)})} \end{aligned}$$

以上用于推导迭代过程中两次L会变大， 这里面红色部分是后加的方便理解前后两步之间的推导。绿色部分是为了构造期望， 进而应用琴声不等式。~~在这里凑项应该是凑~~$P(Z|Y,\theta^{(i)})$~~，书中这部分可能是笔误~~(第二版已经修正)。

这里也写一下琴声不等式

$$\log \sum_j \lambda_j y_j \geq \sum_j \lambda_j \log y_j, s.t., \lambda_j \geq 0, \sum_j \lambda_j = 1$$

所以，这里的这一项不是随便凑的。

TODO: 更新下这个图9.1，EM算法的解释。

高斯混合模型

混合模型，有多种，高斯混合模型是最常用的。

高斯混合模型(Gaussian Mixture Model)是具有如下**概率分布**的模型:

$$P(y|\theta) = \sum_{k=1}^{K} \alpha_k \phi(y|\theta_k)$$

其中，α_k是系数，$\alpha_k \geq 0$，$\sum\limits_{k=1}^{K} \alpha_k = 1$, $\phi(y|\theta_k)$ 是**高斯分布密度**，$\theta_k = (\mu, \sigma^2)$

$$\phi(y|\theta_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(y-\mu_k)^2}{2\sigma_k^2}\right)$$

上式表示第k个**分**模型。

以上, 注意几点:

1. GMM的描述是概率分布，形式上可以看成是加权求和，有啥用?
2. 加权求和的权重α满足$\sum_{k=1}^{K} \alpha_k = 1$的约束
3. 求和符号中除去权重的部分，是高斯分布密度(PDF)。高斯混合模型是一种$\sum(\text{权重} \times \text{分布密度}) = \text{分布}$的表达
   高斯混合模型的参数估计是EM算法的一个重要应用，隐马尔科夫模型的非监督学习也是EM算法的一个重要应用。
4. 书中描述的是一维的高斯混合模型，d维的形式如下[3]，被称作多元正态分布，也叫多元高斯分布

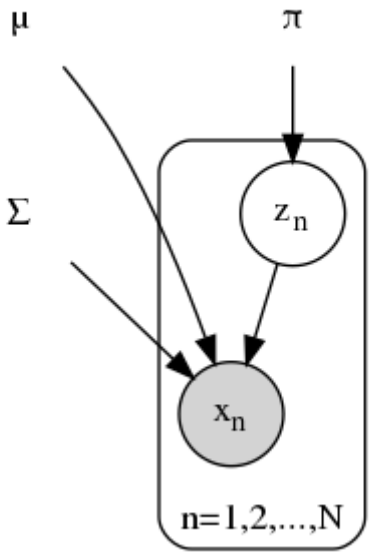
$$\phi(y|\theta_k) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{(y-\mu_k)^T \Sigma^{-1} (y-\mu_k)}{2}\right)$$

其中，协方差矩阵$\Sigma \in \mathbb{R}^{n \times n}$

5. 另外，关于高斯模型的混合，还有另外一种混合的方式，沿着时间轴的方向做混合。可以理解为滤波器，典型的算法就是Kalman Filter，对应了时域与频域之间的关系，两个高斯的混合依然是高斯，混合的方法是卷积，而不是简单的加法，考虑到的是概率密度的混合，也是一种线性的加权。

GMM的图模型

这个弄的不咋好看，plate notation



GMM的EM算法

问题描述:

已知观测数据$y_1, y_2, \ldots, y_N$，由高斯混合模型生成

$$P(y|\theta) = \sum_{k=1}^{K} \alpha_k \phi(y|\theta_k)$$

其中， $\theta = (\alpha_1, \alpha_2, \ldots, \alpha_K; \theta_1, \theta_2, \ldots, \theta_K)$

补充下，不完全数据的似然函数应该是

$$\begin{aligned} P(y|\theta) &= \prod_{j=1}^{N} P(y_j|\theta) \\ &= \prod_{j=1}^{N} \sum_{k=1}^{K} \alpha_k \phi(y|\theta_k) \end{aligned}$$

使用EM算法估计GMM的参数θ

1. 明确隐变量, 初值

- 观测数据$y_j, j = 1, 2, \ldots, N$这样产生, 是**已知的**:
    1. 依概率α_k**选择第**k**个**高斯分布分模型$\phi(y|\theta_k)$;
    2. 依第k个分模型的概率分布$\phi(y|\theta_k)$生成观测数据y_j
    3. 反映观测数据y_j来自第k个分模型的数据是**未知的**, $k = 1, 2, \ldots, K$ 以**隐变量**γ_{jk}表示
    注意这里γ_{jk}**的维度**$(j \times k)$

$$\gamma_{jk} = \begin{cases} 1, & \text{第}j\text{个观测来自第}k\text{个分模型} \\ 0, & \text{否则} \end{cases}$$
$$j = 1, 2, \ldots, N; k = 1, 2, \ldots, K; \gamma_{jk} \in \{0, 1\}$$

注意, 以上说明有几个假设:

1. 隐变量和观测变量的数据对应, 每个观测数据，对应了一个隐变量，γ_{jk}是一种one-hot的形式。
2. 具体的单一观测数据是混合模型中的某一个模型产生的

- 完全数据为$(y_j, \gamma_{j1}, \gamma_{j2}, \ldots, \gamma_{jK}, k = 1, 2, \ldots, N)$
- 完全数据似然函数

$$
\begin{aligned}
P(y,\gamma|\theta) &= \prod_{j=1}^{N} P(y_j,\gamma_{j1},\gamma_{j2},\ldots,\gamma_{jK}|\theta) \\
&= \prod_{k=1}^{K}\prod_{j=1}^{N}[\alpha_k\phi(y_j|\theta_k)]^{\gamma_{jk}} \\
&= \prod_{k=1}^{K}\alpha_k^{n_k}\prod_{j=1}^{N}[\phi(y_j|\theta_k)]^{\gamma_{jk}} \\
&= \prod_{k=1}^{K}\alpha_k^{n_k}\prod_{j=1}^{N}\left[\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{(y_j-\mu_k)^2}{2\sigma^2}\right)\right]^{\gamma_{jk}}
\end{aligned}
$$

其中$n_k=\sum_{j=1}^{N}\gamma_{jk}, \sum_{k=1}^{K}n_k=N$

- 完全数据对数似然函数

$$
\log P(y,\gamma|\theta)=\sum_{k=1}^{K}\left\{n_k\log\alpha_k+\sum_{j=1}^{N}\gamma_{jk}\left[\log\left(\frac{1}{\sqrt{2\pi}}\right)-\log\sigma_k-\frac{1}{2\sigma^2}(y_j-\mu_k)^2\right]\right\}
$$

2. E步,确定Q函数

把Q 函数表示成参数形式

$$
\begin{aligned}
Q(\theta,\theta^{(i)}) =& E[\log P(y,\gamma|\theta)|y,\theta^{(i)}] \\
=& E\sum_{k=1}^{K}n_k\log\alpha_k+\sum_{j=1}^{N}\gamma_{jk}[\log(\frac{1}{\sqrt{2\pi}})-\log\sigma_k-\frac{1}{2\sigma^2(y_j-\mu_k)^2}] \\
=& E\sum_{k=1}^{K}\sum_{j=1}^{N}\gamma_{jk}\log\alpha_k+\sum_{j=1}^{N}\gamma_{jk}[\log(\frac{1}{\sqrt{2\pi}})-\log\sigma_k-\frac{1}{2\sigma^2(y_j-\mu_k)^2}] \\
=& \sum_{k=1}^{K}\sum_{j=1}^{N}(E\gamma_{jk})\log\alpha_k+\sum_{j=1}^{N}(E\gamma_{jk})[\log(\frac{1}{\sqrt{2\pi}})-\log\sigma_k-\frac{1}{2\sigma^2(y_j-\mu_k)^2}]
\end{aligned}
$$

$$
\begin{aligned}
\hat{\gamma}_{jk} =& E(\gamma_{jk}|y,\theta)=P(\gamma_{jk}=1|y,\theta) \\
=& \frac{P(\gamma_{jk}=1,y_j|\theta)}{\sum_{k=1}^{K}P(\gamma_{jk}=1,y_j|\theta)} \\
=& \frac{P(y_j|\gamma_{jk}=1,\theta)P(\gamma_{jk}=1|\theta)}{\sum_{k=1}^{K}P(y_j|\gamma_{jk}=1,\theta)P(\gamma_{jk}=1|\theta)} \\
=& \frac{\alpha_k\phi(y_j|\theta_k)}{\sum_{k=1}^{K}\alpha_k\phi(y_j|\theta_k)}
\end{aligned}
$$

这部分内容就是搬运了书上的公式，有几点说明:

1. 注意这里$E(\gamma_{jk}|y,\theta)$，记为$\hat{\gamma}_{jk}$， 对应了E步求的**期望**中的一部分。
2. 对应理解一下上面公式中的红色，蓝色和绿色部分，以及$\hat{\gamma}_{jk}$中红色和绿色的对应关系
3. 这里用到了$n_k=\sum_{j=1}^{N}\gamma_{jk}$
4. $\hat{\gamma}_{jk}$为分模型k对观测数据y_j的响应度。这里，紫色标记的第一行参考伯努利分布的期望。

$$
Q(\theta,\theta^{(i)})=\sum_{k=1}^{K}n_k\log\alpha_k+\sum_{j=1}^{N}\hat{\gamma}_{jk}\left[\log\left(\frac{1}{\sqrt{2\pi}}\right)-\log\sigma_k-\frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]
$$

其中i表示第i步迭代

1. 写出Q 函数在推导的时候有用，但是在程序计算的时候，E步需要计算的就是$\hat{\gamma}_{jk}$，M步用到了这个结果。其实抄公式没有什么意义，主要是能放慢看公式的速度。和图表一样，公式简洁的表达了很多信息，公式中也许更能体会到数学之美。

3. M步

求函数$Q(\theta,\theta^{(i)})$对θ的极大值，分别求σ,μ,α

$$
\theta^{(i+1)}=\arg\max_{\theta}Q(\theta,\theta^{(i)})
$$

- $\arg\max$ 就是求Q的极值对应的参数θ，如说是离散的，遍历所有值，最大查找，如果是连续的，偏导为零求极值。
- $\frac{\partial Q}{\partial \mu_k}=0, \frac{\partial Q}{\partial \sigma^2}=0$ 得到$\hat{\mu}_k, \hat{\sigma}_k^2$
- $\sum_{k=1}^{K} \alpha_k = 1, \frac{\partial Q}{\partial \alpha_k}=0$ 得到α_k

4. 停止条件

重复以上计算，直到对数似然函数值不再有明显的变化为止。

如何应用

GMM在聚类中的应用

使用EM算法估计了GMM的参数之后，有新的数据点，怎么计算样本的类别的?

在机器学习[4]中有一些关于聚类的表述，摘录这里：

> Gaussian mixture modeling is among the popular clustering algorithms. The main assumption is that the points, which belong to the same cluster, are distributed according to the same Gaussian distribution(this is how similarity is defined in this case), of unknown mean and covariance matrix.
>
> Each mixture component defines a different cluster. Thus, the goal is to run the EM algorithm over the available data points to provide, after convergence, the posterior probabilities $P(k|x_n), k=1,2,\ldots,K, n=1,2,\ldots,N$, where each k corresponds to a cluster. Then each point is assigned to cluster k according to the rule.
>
> Assign $x_n$to cluster $k=\arg max P(i|x_n), i=1,2,\ldots,K$

可以参考下scikit-learn的具体实现，就是用了argmax，选择概率最大的那个输出。

这里面，没有介绍如何评价拟合之后的一个模型，文章Measuring the component overlapping in the Gaussian mixture model中，有描述OLR的方法，可以参考。

这里还有个实现。

GMM在CV中的应用

其实CV中用到了很多统计的方法，GMM在GrabCut方法中用于前景和背景建模。

算法9.2

这部分摘要总结了前面的公式。

因为公式比较集中，方便对比，注意体会以下两个方面:

1. 这几个公式中待求的变量的维度和角标的关系。
2. 这里面有求和，前面提到过要注意体会每一步刷的是模型，还是样本

Kmeans

另外，直觉上看，GMM最直观的想法就是Kmeans，那么:

1. 在Kmeans常见的描述中都有距离的概念，对应在算法9.2 的描述中，该如何理解?
   这里面距离对应了方差，二范数平方。
2. 那么又是怎么在每轮刷过距离之后，重新划分样本的分类呢?
   这里对应了响应度，响应度对应了一个$j \times k$的矩阵，记录了每一个y_j 对第k个模型的响应度，可以理解为划分了类别。

K怎么定

- 手肘法
- Gap Statistics[5]
- 第二版中，有对应的描述。P_{267}，一般的类别变大的时候，平均直径会增加，当类别数超过某个值之后，平均直径不会变化，这个值可以是最优的k值。

广义期望极大

广义期望极大(generalized expectation maximization, GEM)

广义期望极大是为了解决什么问题?

看名字是为了通用解决方案吧

其他

习题9.3

GMM模型的参数($\alpha_k, \mu_k, \sigma_k^2$)应该是$3k$个，题目9.3中提出两个分量的高斯混合模型的5个参数，是因为参数α_k满足$\sum_{k=1}^{K} \alpha_k = 1$

习题9.4

EM算法用到朴素贝叶斯的非监督学习，就是说没有标注的数据。

这个题目可以参考https://ttic.uchicago.edu/~suriya/website-intromlss2018/course_material/Day10a.pdf

参考

1.
2. EM Algorithm
3. Sklearn Gaussian Mixed Model
4.
5.
6. mml
7.
8.
9.

↑ top

#CH10 隐马尔科夫模型

前言

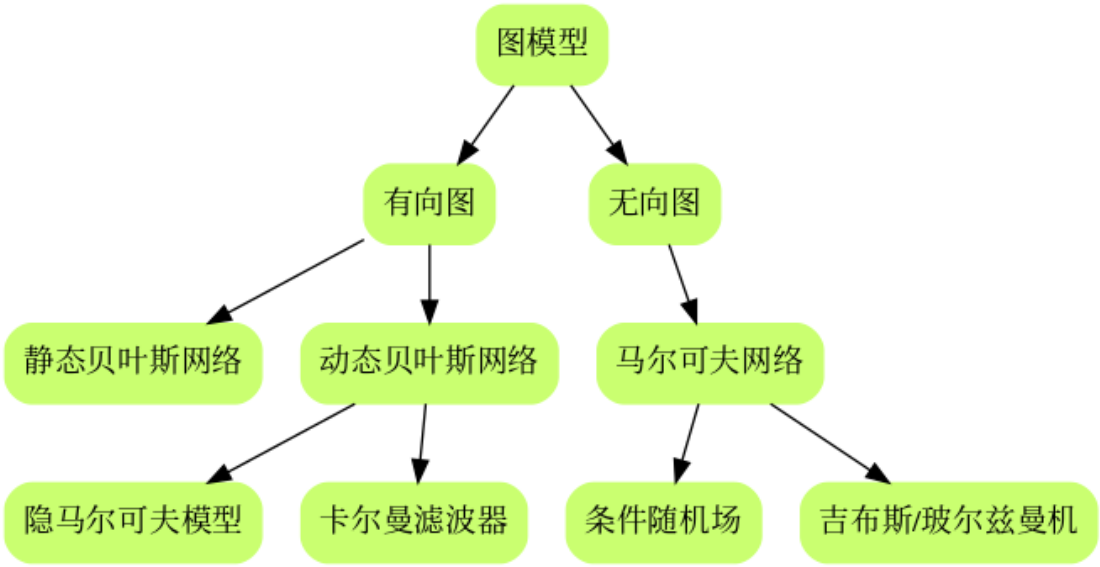
章节目录

1. 隐马尔可夫模型的基本概念
    1. 隐马尔可夫模型的定义
    2. 观测序列的生成过程
    3. 隐马尔可夫模型的三个基本问题
2. 概率计算方法
    1. 直接计算法
    2. 前向算法
    3. 后向算法
    4. 一些概率与期望值的计算

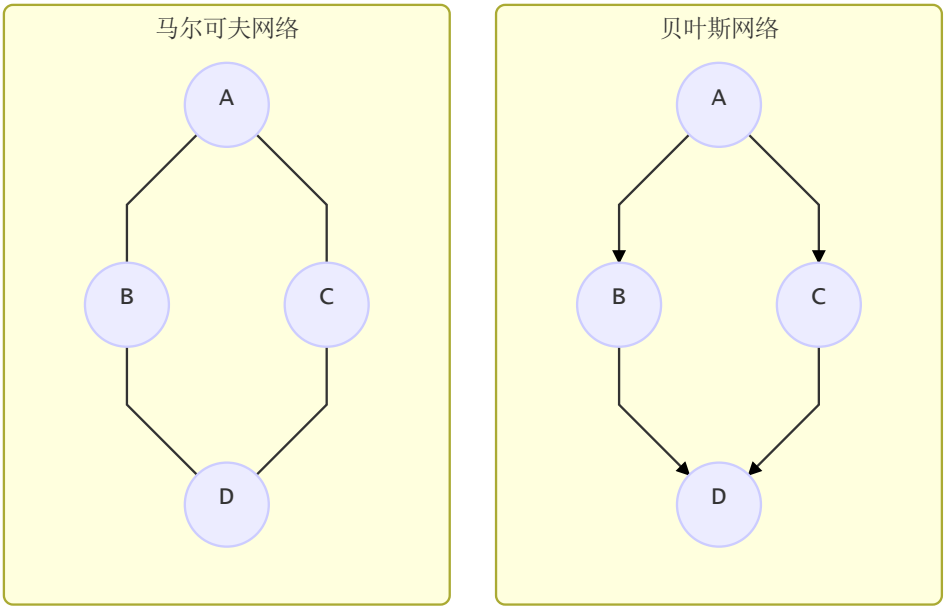
3. 学习算法
    1. 监督学习方法
    2. Baum-Welch算法
    3. Baum-Welch模型参数估计公式
4. 预测算法
    1. 近似算法
    2. 维特比算法

导读

- 我记得在第六章有一个经典的文献介绍最大熵的原理的，例子是语言翻译。这章有个类似的文献就是书中给出的前两个参考文献[2]， 书中的符号体系和书中的参考文献1的保持一致。
- 动态贝叶斯网络的最简单实现隐马尔可夫模型。HMM可以看成是一种推广的混合模型。
- 序列化建模，打破了数据独立同分布的假设。
- 有些关系需要理清



- 另外一个图



另外，注意一点，在李老师这本书上介绍的HMM，涉及到举例子的，给的都是观测概率矩阵是离散的情况，对应了Multinominal HMM。但这个观测概率矩阵是可以为连续的分布的，比如高斯模型，对应了Gaussian HMM，高斯无处不在。具体可以参考hmmlearn库[3]

- HMM有两个基本假设和三个基本问题， 两个基本假设。I是隐变量。
- 才发现这一章居然都没有提到概率图模型。

概念

有些基本的概念， 引用吴军在数学之美[5]之中的描述。

随机变量与随机过程

> 19世纪, 概率论的发展从对(相对静态的)**随机变量**的研究发展到对随机变量的时间序列$s_1, s_2, s_3, \ldots, s_t, \ldots$,即**随机过程**(动态的)的研究
>
> 数学之美,吴军

马尔可夫链

> 随机过程有两个维度的不确定性。马尔可夫为了简化问题，提出了一种简化的假设，即随机过程中各个状态s_t的概率分布，只与它的前一个状态s_{t-1}有关, 即$P(s_t|s_1, s_2, s_3, \ldots, s_{t-1}) = P(s_t|s_{t-1})$
>
> 这个假设后来被称为**马尔可夫假设**，而符合这个假设的随机过程则称为**马尔可夫过程**，也称为**马尔可夫链**。
>
> 数学之美，吴军

$$P(s_t|s_1, s_2, s_3, \ldots, s_{t-1}) = P(s_t|s_{t-1})$$

时间和状态取值都是离散的马尔可夫过程也称为马尔可夫链。

隐含马尔可夫模型

$$P(s_1, s_2, s_3, \ldots, o_1, o_2, o_3, \ldots) = \prod_t P(s_t|s_{t-1}) \cdot P(o_t|s_t)$$

隐含的是**状态**s

隐含马尔可夫模型由**初始概率分布**(向量π), **状态转移概率分布**(矩阵A)以及**观测概率分布**(矩阵B)确定.

隐马尔可夫模型λ 可以用三元符号表示, 即

$$\lambda = (A, B, \pi)$$

其中A, B, π称为模型三要素。

具体实现的过程中，如果观测的概率分布是定的，那么B就是确定的。在hhmlearn[3]中，实现了三种概率分布的HMM模型：MultinominalHMM，GaussianHMM，GMMHMM。还可以定义不同的emission probabilities[1]，生成不同的HMM模型。

两个基本假设

1. 齐次马尔科夫假设(**状态**)

   $$P(i_t|i_{t-1}, o_{t-1}, \ldots, i_1, o_1) = P(i_t|i_{t-1}), t = 1, 2, \ldots, T$$

   注意书里这部分的描述

   > 假设隐藏的马尔可夫链在**任意时刻**t**的状态**$\rightarrow i_t$
   > 只依赖于其前一时刻的状态$\rightarrow i_{t-1}$
   > 与其他时刻的状态 $\rightarrow i_{t-1,\ldots,i_1}$
   > 及观测无关 $\rightarrow o_{t-1}, \ldots, o_1$
   > 也与时刻t无关 $\rightarrow t = 1, 2, \ldots, T$

   如此烦绕的一句话, 用一个公式就表示了, 数学是如此美妙.

2. 观测独立性假设(**观测**)

   $$P(o_t|i_T, o_T, i_{T-1}, o_{T-1}, \ldots, i_{t+1}, o_{t+1}, i_t, i_{t-1}, o_{t-1}, \ldots, i_1, o_1) = P(o_t|i_t)$$

   书里这部分描述如下

   > 假设**任意时刻**t**的观测**$\rightarrow o_t$
   > 只依赖于该时刻的马尔可夫链的状态 $\rightarrow i_t$
   > 与其他观测 $\rightarrow o_T, o_{T-1}, \ldots, o_{t+1}, o_{t-1}, \ldots, o_1$
   > 及状态无关 $\rightarrow i_T, i_{T-1}, \ldots, i_{t+1}, i_{t-1}, \ldots, i_1$

   李老师这个书真的是无废话

三个基本问题

1. 概率计算问题
   输入: 模型$\lambda = (A, B, \pi)$, 观测序列$O = (o_1, o_2, \ldots, o_T)$
   输出: $P(O|\lambda)$
2. 学习问题
   输入: 观测序列 $O = (o_1, o_2, \ldots, o_T)$
   输出: 输出$\lambda = (A, B, \pi)$
3. 预测问题, 也称为解码问题(Decoding)
   输入: 模型$\lambda = (A, B, \pi)$, 观测序列$O = (o_1, o_2, \ldots, o_T)$
   输出: 状态序列 $I = (i_1, i_2, \ldots, i_T)$

   因为状态序列是隐藏的，不可观测的，所以叫解码。

> There are three fundamental problems for HMMs:

> - Given the model parameters and observed data, estimate the optimal sequence of hidden states.
> - Given the model parameters and observed data, calculate the likelihood of the data.
> - Given just the observed data, estimate the model parameters.
>
> The first and the second problem can be solved by the dynamic programming algorithms known as the Viterbi algorithm and the Forward-Backward algorithm, respectively. The last one can be solved by an iterative Expectation-Maximization (EM) algorithm, known as the Baum-Welch algorithm.
>
> ---hhmlearn

算法

观测序列生成算法

> 输入：$\lambda=(A,B,\pi)$,观测序列长度T
>
> 输出：观测序列$O=(o_1,o_2,\dots,o_T)$
>
> 1. 按照初始状态分布π产生i_1
> 2. $t=1$
> 3. 按照状态i_t的观测概率分布$b_{i_t}(k)$生成o_t
> 4. 按照状态i_t的状态转移概率分布$\{a_{i_t,i_{t+1}}\}$产生状态i_{t+1},$i_{t+1}=1,2,\dots,N$
> 5. $t=t+1$ 如果$t<T$转到3,否则，终止

上面是书中的描述，和本章大参考文献[4]的描述是一样的， 但这里面有点容易混淆。

书中定义了$I=(i_1,i_2,\dots,i_T),Q=\{q_1,q_2,\dots,q_T\}$根据定义， i_t的取值集合应该是Q，而上面算法描述中说明了$i_{t+1}=1,2,\dots,N$
注意这里面的i_t实际上不是状态， 而是对应了前面的i,j的含义，实际的状态应该是q_{i_t}这个算法中的$a_{i_ti_{t+1}}=P(i_{t+1}=q_{i_{t+1}}|i_t=q_{i_t})$ 这里同样的符号，表示了两个不同的含义。

Rabiner定义的a_{ij}是这样的

$$A=a_{ij},a_{ij}=Pr(q_jatt+1|q_iatt)$$

这里理解就好， 有时候用角标i代表对应的state， 有时候用q_i代表对应的state。

学习算法

概率计算算法

前向概率与后向概率

> 给定马尔可夫模型λ, 定义到时刻t部分观测序列为$o_1,o_2,\dots,o_t$, 且状态q_i的概率为**前向概率**, 记作
>
> $$\alpha_t(i)=P(o_1,o_2,\dots,o_t,i_t=q_i|\lambda)$$
>
> 给定马尔可夫模型λ, 定义到时刻t状态为q_i的条件下, 从$t+1$到T的部分观测序列为$o_{t+1},o_{t+2},\dots,o_T$的概率为**后向概率**, 记作
>
> $$\beta_t(i)=P(o_{t+1},o_{t+2},\dots,o_T|i_t=q_i,\lambda)$$
>
> 关于α和β这两个公式, 仔细看下, 细心理解. 前向概率从前往后递推， 后向概率从后向前递推。

前向算法

> 输入: λ,O
>
> 输出:$P(O|\lambda)$
>
> 1. 初值

$$\alpha_1(i) = \pi_i b_i(o_1), i = 1, 2, \dots, N$$

观测值o_1, i的含义是对应状态q_i

这里α是N维向量, 和状态集合Q的大小N有关系. α是个联合概率.

2. 递推

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(o_{t+1}),\ i = 1, 2, \dots, N,\ t = 1, 2, \dots, T-1$$

转移矩阵A维度$N \times N$, 观测矩阵B维度$N \times M$, 具体的观测值o可以表示成one-hot形式, 维度$M \times 1$, 所以α的维度是$\alpha = \alpha ABo = 1 \times N \times N \times N \times N \times M \times M \times N = 1 \times N$

3. 终止

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} \alpha_T(i)\beta_T(i)$$

注意, 这里$O \to (o_1, o_2, o_3, \dots, o_t)$, α_i见前面前向概率的定义$P(o_1, o_2, \dots, o_t, i_t = q_i|\lambda)$, 所以, 对$i$求和能把联合概率中的$I$消掉.

这个书里面解释的部分有说.

> 书中有说前向算法的关键是其局部计算前向概率, 然后利用路径结构将前向概率"递推"到全局.
>
> 减少计算量的原因在于每一次计算直接引用前一时刻的计算结果, **避免重复计算**.

前向算法计算$P(O|\lambda)$的复杂度是$O(N^2T)$阶的，直接计算的复杂度是$O(TN^T)$阶，所以$T = 2$时候并没什么改善。

红色部分为后补充了$\beta_T(i)$项，这项为1,此处注意和后面的后向概率对比。

后向算法

输入: λ, O
输出:$P(O|\lambda)$

1. 终值

$$\beta_T(i) = 1, i = 1, 2, \dots, N$$

在$t = T$时刻, 观测序列已经确定.

2. 递推

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), i = 1, 2, \dots, N, t = T-1, T-2, \dots, 1$$

从后往前推
$\beta = ABo\beta = N \times N \times N \times M \times M \times N \times N \times 1 = N \times 1$

3.

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i) = \sum_{i=1} \alpha_1(i)\beta_1(i)$$

- 这里需要**注意**下，按照后向算法，β在递推过程中会越来越小，如果层数较多，怕是$P(O|\lambda)$会消失
- 另外一个要注意的点$o_{t+1}\beta_{t+1}$
- 注意，红色部分为后补充，结合前面的前向概率最后的红色部分一起理解。

小结

> 求解的都是**观测序列概率**
> 观测序列概率$P(O|\lambda)$统一写成
>
> $$P(O|\lambda)=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1}\beta_{t+1}(j)),\ t=1,2,\dots,T-1$$
>
> $P(O|\lambda)=\alpha ABo\beta$

其实前向和后向不是为了求整个序列O的概率，是为了求中间的某个点t，前向后向主要是有这个关系:

$$\alpha_t(i)\beta_t(i)=P(i_t=q_i,O|\lambda)$$

当$t=1$或者$t=T-1$的时候，单独用后向和前向就可以求得$P(O|\lambda)$，分别利用前向和后向算法均可以求解$P(O|\lambda)$，结果一致。

利用上述关系可以得到下面一些概率和期望，这些概率和期望的表达式在后面估计模型参数的时候有应用。

概率与期望

1. 输入模型λ与观测O，输出在时刻t处于状态q_i的概率$\gamma_t(i)$
2. 输入模型λ与观测O，输出在时刻t处于状态q_i且在时刻$t+1$处于状态q_j的概率$\xi_t(i,j)$
3. 在观测O下状态i出现的期望值
4. 在观测O下状态i转移的期望值
5. 在观测O下状态i转移到状态j的期望值

监督学习方法

效果好，费钱，如果有钱能拿到标注数据，不用犹豫，去干吧。

Baum-Welch算法

马尔可夫模型实际上是一个含有隐变量的概率模型

$$P(O|\lambda)=\sum_I P(O|I,\lambda)P(I|\lambda)$$

关于EM算法可以参考第九章, 对隐变量求期望, Q函数极大化

> 输入: 观测数据$O=(o_1,o_2,\dots,o_T)$
>
> 输出: 隐马尔可夫模型参数
>
> 1. 初始化
>   对$n=0$，选取$a_{ij}^{(0)},b_j(k)^{(0)},\pi_i^{(0)}$，得到模型参数$\lambda^{(0)}=(A^{(0)},B^{(0)},\pi^{(0)})$
> 2. 递推
>   对$n=1,2,\dots,$
>
> $$a_{ij}^{(n+1)}=\frac{\sum\limits_{t=1}^{T-1}\xi_t(i,j)}{\sum\limits_{t=1}^{T-1}\gamma_t(i)}$$
>
> $$b_j(k)^{(n+1)}=\frac{\sum\limits_{t=1,o_t=v_k}^{T}\gamma_t(j)}{\sum\limits_{t=1}^{T}\gamma_t(j)}$$
>
> $$\pi_i^{(n+1)}=\gamma_1(i)$$
>
> 1. 终止
>   得到模型参数$\lambda^{(n+1)}=(A^{(n+1)},B^{(n+1)},\pi^{(n+1)})$

$b_j(k)$的理解

单独说一下这个问题，公式里面求和有个$o_t = v_k$, 什么意思?

γ的维度应该是$N \times T$，通过$\sum_{t=1}^{T}$可以降维到N，但是实际上B的维度是$N \times M$，所以有了这个表达，~~窃以为这里可以表示成~~b_{jk}，书中对应部分的表达在P_{172}的10.3，也说明了$b_j(k)$的具体定义。

注意这里$b_j(k)$并不要求是离散的，可以定义为一个连续的函数， 所以书中这样的表达更通用一些，关于这点在本章大参考文献[4]中有部分内容讨论，见`Special cases of the B parameters`。

这里涉及到实际实现的时候，可以考虑把观测序列O转换成one-hot的形式, O_{one_hot}维度为$M \times T$，B的维度$N \times M$，$B \cdot O$之后，转换成观测序列对应的发射概率矩阵，维度为$N \times T$。

补充一下， $o_t = v_k$有另外一种表达是σ_{o_t,v_k}， 克罗内克函数。

克罗内克函数是一个二元函数， 自变量一般是两个整数， 如果两者相等， 输出是1, 否则为0.

其实和指示函数差不多， 只不过条件只限制在了相等。

$$\sigma_{ij} = \begin{cases} 1(i=j) \\ 0(i \neq j) \end{cases}$$

$$b_j(k) = \frac{\sum\limits_{t=1,o_t=v_k}^{T} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)} = \frac{\sum\limits_{t=1}^{T} \sigma_{o_t,v_k} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)}$$

E步与M步的理解

Baum-Welch算法是EM算法在隐马尔可夫模型学习中的**具体实现**, 由Baum和Welch提出.

看到书上这里都知道是EM算法, 具体实现哪里是E, 哪里是M?

书中在前向后向算法介绍之后, 单独有一个小节介绍了"一些概率与**期望**值的计算", 这部分内容在后面的Baum-Welch算法中会用到, 代码实现的时候才理解, 这小节对应的是E步概率和期望, 后面算法里面的是M步的内容, 说明如何用这些概率和期望去更新HMM模型的参数.

重新梳理一下整个10.2节的内容，这部分内容描述**概率计算方法** ， 实际上在E步操作的时候都要用到，需要用到前向后向算法根据模型参数A, B, π来更新α和β，然后利用这两个值来更新一些概率和期望，再通过模型参数的递推公式来更新模型参数。

这里可能还有点疑问，EM算法的描述里面，E步计算的是Q函数，但是前面的描述似乎并没有显示Q函数和这些工作之间的关系。另外，M步具体操作是参数更新的递推公式，怎么就是最大化了呢? 书中P_{182}的推导也许能解释这个问题。

看到这里, 感觉书上真的是一句废话都没有...

这部分的理解，要再结合第九章的内容反复一下，应该会有新的体会。

注意E步计算Q函数

$$Q(\lambda, \bar{\lambda}) = \sum_{I} \log P(O, I|\lambda) P(O, I|\bar{\lambda})$$

对比一下算法9.1，$P(O, I|\bar{\lambda}) = P(I|O, \bar{\lambda}) P(O|\bar{\lambda})$，所以书中在这个地方有个注释，略去了对于$\lambda$而言的常数因子$1/P(O|\bar{\lambda})$

预测算法

近似算法(MAP)

每个时刻最有可能的状态i_t^*是

$$i_t^* = \arg \max_{1 \leqslant i \leqslant N} \left[\gamma_t(i) \right], t = 1, 2, \ldots, T$$

得到序列$I^* = (i_1^*, i_2^*, \dots, i_T^*)$

这个算法, 在输出每个状态的时候, 只考虑了当前的状态.

维特比算法(Viterbi)

> 输入: 模型$\lambda = (A, B, \pi)$和观测$O = (o_1, o_2, \dots, o_T)$
>
> 输出: 最优路径$I^* = (i_1^*, i_2^*, \dots, i_T^*)$
>
> 1. 初始化
>
> $\delta_1(i) = \pi_i b_i(o_1), i = 1, 2, \dots, N$
>
> $\psi_1(i) = 0, i = 1, 2, \dots, N$
>
> 2. 递推
>
> $t = 2, 3, \dots, T$
>
> $\delta_t(i) = \max\limits_{1 \leqslant j \leqslant N} \left[\delta_{t-1}(j) a_{ji}\right] b_i(o_t), i = 1, 2, \dots, N$
>
> $\psi_t(j) = \arg \max\limits_{1 \leqslant j \leqslant N} \left[\delta_{t-1}(j) a_{ji}\right], i = 1, 2, \dots, N$
>
> 3. 终止
>
> $P^* = \max\limits_{1 \leqslant i \leqslant N} \delta_T(i)$
>
> $i_T^* = \arg \max\limits_{1 \leqslant i \leqslant N} \left[\delta_T(i)\right]$
>
> 4. 最优路径回溯
>
> $t = T-1, T-2, \dots, 1$
>
> $i_t^* = \psi_{t+1}(i_{i+1}^*)$

书上配了个图，这个图可视化了δ

例子

例10.1

这个例子主要说明怎么从数据中拿到状态集合, 观测集合, 序列长度以及模型三要素.

分清楚哪些是已知, 哪些是推导得到.

书中描述也很清楚

> 这是一个隐马尔可夫模型的例子, 根据所给条件, 可以明确状态集合, 观测集合, 序列长度以及模型三要素.

恩, 例子干的就是这个事, 而这个小节叫做 隐马尔可夫模型的定义.

例10.2

这个例子就是递推的~~~~矩阵乘法~~

$$\alpha = ABO$$

因为是递推的, 所以没办法用矩阵乘法实现.

这里针对这个例子说下自己的看法, 这个例子稍微有点特殊T和N都是3, 这种情况对展开分析算法不是很合适, 如果$T = 4$有些问题可能会更容易分析.

例10.3

求最优状态序列

这个例子相对简单了, 在验证的过程中, 要核对δ的结果.

上面两个例子真的比较特殊,~~状态转移矩阵还是对称矩阵~~.

状态转移矩阵A肯定是个方阵，但是这个例子里面状态数和序列长度一样， 稍微有点不方便。

习题

前面在处理例10.2的时候, 还觉得这个例子不合适. 翻看后面习题的过程中发现, 有些点是在习题中有所展开.

习题10.1

模型参数和例子10.2是一样的, 只是改变了观测序列长度.

习题 10.2

状态转移矩阵非对称, 观测序列长度更长, 求解的过程需要使用前向后向一起求解.

针对这个问题$i_4 = q_3$给定了条件$t = 4, i = 3$, 那么公式10.22有下面的形式

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_4(3) a_{3j} b_j(o_5) \beta_5(j)$$

这个习题应该是要思考观测序列概率的形式吧, 应该是一个维度是$N \times N \times T$的一个三维矩阵.

习题 10.3

求例子10.1 的隐状态序列

习题10.4

习题10.5

这个在自己推导的过程中, 不自觉的注意到了.

推荐按照书中的例子, 推导α, β, δ, 在推导的过程中, 会发现α, δ有相同的初值. 当然, 这个公式定义上也是一样的.

一个用到了求和, 另外一个用到了求最大值.

实际问题

手写数字生成

采样应用

隐马尔可夫模型的一个强大的性质是他对与时间轴上的局部的变形具有某种程度的不变性.

中文分词

有几个问题要弄清:

1. 怎么评价分词效果的好坏?
2. 模型参数训练的过程, 迭代应该在什么时候停止?

参考

1.
2.
3.
4.
5. Wikipedia: Hidden Markov Model

6.

↑ top

CH11 条件随机场

前言

章节目录

导读

- 条件随机场是给定一组输入随机变量的**条件**下另一组输出随机变量的条件概率分布模型，其特点是假设输出随机变量构成马尔可夫**随机场**。注意这里条件，随机场的对应。
- 整个这一章的介绍思路，和前一章有点像，尤其是学习算法部分，和HMM比主要增加了特征函数，关于特征函数要和CH06对比着看
- CRF是对数线性模型
- 概率无向图模型又称马尔可夫随机场，是可以用无向图表示的**联合概率分布**，注意，一个概率图模型就是一个联合概率分布。
- 条件随机场三个基本问题：概率计算问题，学习问题和预测问题
- 前面的章节中，我们学习，更新参数的过程中很多时候用到了导数，那么积分有没有用？联合概率分布包含了很多随机变量，如果希望消除一些变量，就会用到积分，在离散的情况下，就是求和。这个过程就是边缘化。所有的概率图模型都尝试提出有效的方法来解决这个积分的问题。
- 统计力学中，波尔兹曼分布是描述粒子处于特定状态下的概率，是关于状态能量与系统温度的函数。
  $P_\alpha = \frac{1}{Z}\exp(\frac{-E_\alpha}{kT})$
  p_α是粒子处于状态α的概率，E_α为状态α的能量，k为波尔兹曼常量，T为系统温度，$\exp(\frac{-E_\alpha}{kT})$称为波尔兹曼因子，是没有归一化的概率，$Z$为归一化因子，是对系统所有状态进行总和。
  在统计力学中， Z一般称为配分函数，其定义为
  $Z = \sum_\alpha \exp(\frac{-E_\alpha}{kT})$
  波尔兹曼分布的一个性质是两个状态的概率的比值，仅仅依赖于两个状态能量的差值，这里除法变减法，想到的应该是指数分布。
  $\frac{p_\alpha}{p_\beta} = \exp(\frac{E_\beta - E_\alpha}{kT})$
  所以，系统应该倾向于停留在能量大的状态。
-

概念

符号表

节点$\nu \in V$表示一个随机变量Y_ν

边$e \in E$表示随机变量之间的概率依赖关系

图$G(V, E)$表示联合概率分布$P(Y)$

$Y \in \mathcal{Y}$是**一组随机变量**$Y = (Y_\nu)_{\nu \in V}$

IOB标记

Inside, **O**utside, **B**egin

概率无向图模型

注意整个书中第一节的内容， 还不是条件随机场， 都是马尔可夫随机场，告诉读者可以用图来表示联合分布，以及拿到图之后， 怎么转化成概率表达形式。

概率无向图模型又称**马尔可夫随机场(MRF)**，是一个可以由**满足以下三个性质的无向图**表示的**联合概率分布。**

- 成对马尔可夫性
  给定随机变量组Y_O的条件下随机变量Y_u和Y_v是条件独立的
  $P(Y_u, Y_v|Y_O) = P(Y_u|Y_O)P(Y_v|Y_O)$
- 局部马尔可夫性
  给定随机变量组Y_W的条件下随机变量Y_v与随机变量组Y_O是独立的
  $P(Y_v, Y_O|Y_W) = P(Y_v|Y_W)P(Y_O|Y_W)$
- 全局马尔可夫性
  给定随机变量组Y_C的条件下随机变量组Y_A和Y_B是条件独立的
  $P(Y_A, Y_B|Y_C) = P(Y_A|Y_C)P(Y_B|Y_C)$

MRF的因子分解

> 将概率无向图模型的联合概率分布表示为其**最大团**上的随机变量的函数的乘积形式的操作，称为概率无向图模型的因子分解(factorization)
> 概率无向图模型的最大特点就是**易于因子分解**

团与最大团

有向图模型

插入一点有向图模型

条件随机场

条件随机场是给定随机变量X条件下，随机变量Y的马尔可夫随机场。

线性链条件随机场

设$X = (X_1, X_2, \cdots, X_n)$, $Y = (Y_1, Y_2, \cdots, Y_n)$均为线性链表示的随机变量序列，若在给定随机变量序列X的条件下，随机变量序列Y的条件概率分布$P(Y|X)$构成条件随机场，即满足马尔可夫性

$$P(Y_i|X, Y_1, \cdots, Y_{i-1}, Y_{i+1}, \cdots, Y_n) = P(Y_i|X, Y_{i-1}, Y_{i+1})$$
$$i = 1, 2, \cdots, n(\text{在}i = 1\text{和}n\text{时只考虑单边})$$

则称$P(Y|X)$为线性链条件随机场。**在标注问题中**，X表示输入观测序列， Y表示输出标记序列或状态序列。

特征函数

线性链条件随机场的参数化形式

$$P(y|x)=\frac{1}{Z(x)}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}\mu_l s_l(y_i,x,i)\right)$$

其中

t_k是定义在**边**上的特征函数，称为转移特征

s_l是定义在**结点**上的特征函数，称为状态特征

注意到这种表达就是不同特征的加权求和形式，t_k,s_l都依赖于位置，是局部特征函数。

对数线性模型

线性链条件随机场也是**对数线性模型**(定义在时序数据上的)。

条件随机场可以看做是最大熵马尔可夫模型在标注问题上的推广。

条件随机场是计算**联合概率分布**的有效模型。

> 现实中，一般假设X和Y有相同的图结构。
>
> 本书主要考虑无向图为
>
> $$G=(V=1,2,\dots,n,E=(i,i+1)),i=1,2,\dots,n-1$$
>
> 在此情况下，$X=(X_1,X_2,\dots,X_n),Y=(Y_1,Y_2,\dots,Y_n)$

线性链条件随机场定义

> 设$X=(X_1,X_2,\dots,X_n),Y=(Y_1,Y_2,\dots,Y_n)$均为线性链表示的随机变量序列， 若在给定随机变量序列X的条件下， 随机变量序列Y的条件概率分布$P(Y|X)$构成条件随机场， 即满足马尔可夫性
> $P(Y_i|X,Y_1,\dots,Y_{i-1},Y_{i+1},\dots,Y_n)=P(Y_i|X,Y_{i-1},Yi+1),i=1,2,\dots,n$
>
> 则称$P(Y|X)$为线性链条件随机场

参数化形式

随机变量X取值为x的**条件**下，随机变量Y取值为y的条件概率具有如下形式：

$$P(y|x)=\frac{1}{Z}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i+\sum_{i,l}\mu_l s_l(y_i,x,i)\right)$$

其中

$$Z(x)=\sum_y\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}\mu_l s_l(y_i,x,i)\right)$$

k,l对应特征函数的编号，注意这里用了k,l两个编号，i对应了输出序列的每个位置

$$\begin{array}{lll}
t_1=t_1(y_{i-1}=1,y_i=2,x,i), & i=2,3, & \lambda_1=1\\
t_2=t_2(y_{i-1}=1,y_i=1,x,i), & i=2, & \lambda_2=0.5\\
t_3=t_3(y_{i-1}=2,y_i=1,x,i), & i=3, & \lambda_3=1\\
t_4=t_4(y_{i-1}=2,y_i=1,x,i), & i=2, & \lambda_4=1\\
t_5=t_5(y_{i-1}=2,y_i=2,x,i), & i=3, & \lambda_5=0.2\\
s_1=s_1(y_i=1,x,i), & i=1, & \mu_1=1\\
s_2=s_2(y_i=1,x,i), & i=1,2, & \mu_2=0.5\\
s_3=s_3(y_i=1,x,i), & i=2,3, & \mu_3=0.8\\
s_4=s_4(y_i=2,x,i), & i=3 & \mu_4=0.5
\end{array}$$

可以抽象成上面这种形式。

简化形式

上面的结构，包含了两个部分，表达式不够简单，如何落地？

K_1个转移特征， K_2个状态特征

$$f_k(y_{i-1},y_i,x,i)=\begin{cases}t_k(y_{i-1},y_i,x,i), & k=1,2,\dots,K_1\\ s_l(y_i,x,i), & k=K_1+l;l=1,2,\dots,K_2\end{cases}$$

上面这个红色的式子很重要，把unigram和bigram统一到一起了，如果有trigram等也在这里融合

然后，对转和状态特征在各个位置i求和，记作

$$f_k(y,x)=\sum_{i=1}^{n}f_k(y_{i-1},y_i,x,i),k=1,2,\dots,K$$

用w_k表示特征$f_k(y,x)$的权值

$$w_k=\begin{cases}\lambda_k, & k=1,2,\dots,K_1\\ \mu_l, & k=K1+l;l=1,2,\dots,K_2\end{cases}$$

于是条件随机场可以表示为

$$P(y|x)=\frac{1}{Z(x)}\exp\sum_{k=1}^{K}w_kf_k(y,x)$$
$$Z(x)=\sum_{y}\exp\sum_{k=1}^{K}w_kf_k(y,x)$$

若以w表示权值向量， 即

$$w=(w_1,w_2,\dots,w_K)^T$$

以F表示全局特征向量，即

$$F(y,x)=(f_1(y,x),f_2(y,x),\dots,f_K(y,x))^T$$

条件随机场可以表示成向量内积的形式

$$P_w(y|x)=\frac{\exp(w\cdot F(y,x))}{Z_w(x)}$$
$$Z_w(x)=\sum_{y}\exp(w\cdot F(y,x))$$

在参数化形式的展示中，书中的公式已经做了删减。
而实际上这里应该是展开的。

$$
\begin{aligned}
&f_k = t_1(y_{i-1}=1, y_i=2, x, i), && i=2, && w_k=1, && k=1\\
&f_k = t_1(y_{i-1}=1, y_i=2, x, i), && i=3, && w_k=1, && k=1\\
&f_k = t_2(y_{i-1}=1, y_i=1, x, i), && i=2, && w_k=0.5, && k=2\\
&f_k = t_2(y_{i-1}=1, y_i=1, x, i), && i=3, && w_k=0.5, && k=2\\
&f_k = t_3(y_{i-1}=2, y_i=1, x, i), && i=2, && w_k=1, && k=3\\
&f_k = t_3(y_{i-1}=2, y_i=1, x, i), && i=3, && w_k=1, && k=3\\
&f_k = t_4(y_{i-1}=2, y_i=1, x, i), && i=2, && w_k=1, && k=4\\
&f_k = t_4(y_{i-1}=2, y_i=1, x, i), && i=3, && w_k=1, && k=4\\
&f_k = t_5(y_{i-1}=2, y_i=2, x, i), && i=2, && w_k=0.2, && k=5\\
&f_k = t_5(y_{i-1}=2, y_i=2, x, i), && i=3, && w_k=0.2, && k=5
\end{aligned}
$$

$$
\begin{aligned}
&f_k = s_1(y_i=1, x, i), && i=1, && w_k=1, && k=6\\
&f_k = s_1(y_i=1, x, i), && i=2, && w_k=1, && k=6\\
&f_k = s_1(y_i=1, x, i), && i=3, && w_k=1, && k=6\\
&f_k = s_2(y_i=1, x, i), && i=1, && w_k=0.5, && k=7\\
&f_k = s_2(y_i=1, x, i), && i=2, && w_k=0.5, && k=7\\
&f_k = s_2(y_i=1, x, i), && i=3, && w_k=0.5, && k=7\\
&f_k = s_3(y_i=1, x, i), && i=1, && w_k=0.8, && k=8\\
&f_k = s_3(y_i=1, x, i), && i=2, && w_k=0.8, && k=8\\
&f_k = s_3(y_i=1, x, i), && i=3, && w_k=0.8, && k=8\\
&f_k = s_4(y_i=2, x, i), && i=1, && w_k=0.5, && k=9\\
&f_k = s_4(y_i=2, x, i), && i=2, && w_k=0.5, && k=9\\
&f_k = s_4(y_i=2, x, i), && i=3, && w_k=0.5, && k=9
\end{aligned}
$$

这里对于w_k的理解再体会下。

矩阵形式

针对线性链条件随机场

引入起点和终点状态标记$y_0 = start, y_{n+1} = end$， 这时$P_w(y|x)$可以矩阵形式表示。

对应观测序列的**每个位置**$i = 1, 2, \ldots, n+1$，定义一个m阶矩阵（m是标记y_i取值的个数）

$$
\begin{aligned}
M_i(x) &= [M_i(y_{i-1}, y_i|x)]\\
M_i(y_{i-1}, y_i) &= \exp(W_i(y_{i-1}, y_i|x))\\
W_i(y_{i-1}, y_i|x) &= \sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i|x)
\end{aligned}
$$

把整个向量乘法按照**观测位置**拆成矩阵形式， 每个观测位置对应一个矩阵

这个过程和CNN中的卷积实际上有点像，这里面卷积模板有两种$k \times 1$和$k \times 2$， 以1和2进行滑窗。

给定观测序列x，相应的标记序列y的非规范化概率可以通过该序列的$n+1$个矩阵适当元素的乘积$\prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|x)$表示。于是

$$
P_w(y|x) = \frac{1}{Z_w(x)} \prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|x)
$$

其中，Z_w为规范化因子，是$n+1$个矩阵的乘积的$(start, stop)$元素：

$$
Z_w(x) = (M_1(x) M_2(x) \ldots M_{n+1}(x))_{start,stop}
$$

这个式子，以及这段内容，注意下。

上面的式子展开一下，得到$n+1$个m阶矩阵

$$
M_i(x) = \left[\exp\left(\sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i|x) \right) \right], i = 1, 2, \ldots, n+1
$$

这里面， 各个位置$(1,2,\dots,n+1)$的**随机矩阵**分别是

$$
\begin{aligned}
M_1(y_0,y_1|x) &= \exp\left(\sum_{k=1}^{K} w_k f_k(y_0,y_1|x)\right) \\
&= \exp(w_1 f_1(y_0,y_1))\exp(w_2 f_2(y_0,y_1))\dots\exp(w_K f_K(y_0,y_1)) \\
M_2(y_1,y_2|x) &= \exp\left(\sum_{k=1}^{K} w_k f_k(y_1,y_2|x)\right) \\
&= \exp(w_1 f_1(y_1,y_2))\exp(w_2 f_2(y_1,y_2))\dots\exp(w_K f_K(y_1,y_2)) \\
M_3(y_2,y_3|x) &= \exp\left(\sum_{k=1}^{K} w_k f_k(y_2,y_3|x)\right) \\
&= \exp(w_1 f_1(y_2,y_3))\exp(w_2 f_2(y_2,y_3))\dots\exp(w_K f_K(y_2,y_3)) \\
M_4(y_3,y_4|x) &= \exp\left(\sum_{k=1}^{K} w_k f_k(y_3,y_4|x)\right) \\
&= \exp(w_1 f_1(y_3,y_4))\exp(w_2 f_2(y_3,y_4))\dots\exp(w_K f_K(y_3,y_4))
\end{aligned}
$$

所以，无论特征有多少个，随机矩阵都是四个$(n+1)$

这里还有个问题， 这个m阶的矩阵是怎么来的？上面这四个表达式每一个都是m阶矩阵么？

这个问题在例子11.2中展开。

概率计算

前向向量$\alpha_i(x)$

1. 初值
   $\alpha_0(y|x) = \begin{cases} 1, & y = start \\ 0, & others \end{cases}$
2. 递推
   $\alpha_i^T(y_i|x) = \alpha_{i-1}^T(y_{i-1}|x)[M_i(y_{i-1},y_i|x)], i = 1,2,\dots,n+1$

$\alpha_i(y_i|x)$表示在位置i的标记是y_i并且到位置i的前部标记序列的非规范化概率，y_i可取的值有m个，所以$\alpha_i(x)$是m维列向量

后向向量$\beta_i(x)$

1. 初值
   $\beta_{n+1}(y_{n+1}|x) = \begin{cases} 1, & y_{n+1} = stop \\ 0, & others \end{cases}$
2. 递推
   $\beta_i(y_i|x) = [M_{i+1}(y_i,y_{i+1}|x)]\beta_{i+1}(y_{i+1}|x), i = 1,2,\dots,n+1$

$\beta_i(y_i|x)$表示在位置i的标记是y_i并且从$i+1$到n的后部标记序列的非规范化概率

$$
Z(x) = \alpha_n^T(x)\cdot 1 = 1^T\cdot\beta_1(x)
$$

预测

例子

> 条件随机场完全由特征函数t_k, s_l和对应的权值λ_k, μ_l确定
> 接下来的三个例子

- 例11.1

已知上述四个参数的情况下，求概率

- 例11.2

  假设了$y_0 = start = 1, y_4 = stop = 1$

  矩阵形式的表示是为了后面的前向后向算法中递推的使用。

- 例11.3
  decode问题实例

例11.1

特征函数部分的内容理解下

这里整理下题目中的特征函数，这里和书上的格式稍有不同，希望用这样的描述能看到这些特征函数中抽象的地方。

$$
\begin{array}{llll}
t_1 = t_1(y_{i-1} = 1, y_i = 2, x, i), & i = 2, 3, & \lambda_1 = 1 \\
t_2 = t_2(y_{i-1} = 1, y_i = 1, x, i), & i = 2, & \lambda_2 = 0.5 \\
t_3 = t_3(y_{i-1} = 2, y_i = 1, x, i), & i = 3, & \lambda_3 = 1 \\
t_4 = t_4(y_{i-1} = 2, y_i = 1, x, i), & i = 2, & \lambda_4 = 1 \\
t_5 = t_5(y_{i-1} = 2, y_i = 2, x, i), & i = 3, & \lambda_5 = 0.2 \\
s_1 = s_1(y_i = 1, x, i), & i = 1, & \mu_1 = 1 \\
s_2 = s_2(y_i = 1, x, i), & i = 1, 2, & \mu_2 = 0.5 \\
s_3 = s_3(y_i = 1, x, i), & i = 2, 3, & \mu_3 = 0.8 \\
s_4 = s_4(y_i = 2, x, i), & i = 3 & \mu_4 = 0.5
\end{array}
$$

注意上面红色标记的t_3, t_4是可以合并的。

```
# transition feature
# i-1, i
f_k[0] = np.sum([1 if tmp[0] == 1 and tmp[1] == 2 else 0 for tmp in list(zip(Y[:-1],
Y[1:]))])
f_k[1] = np.sum([1 if tmp[0] == 1 and tmp[1] == 1 else 0 for tmp in list(zip(Y[:-1],
Y[1:]))])
f_k[2] = np.sum([1 if tmp[0] == 2 and tmp[1] == 1 else 0 for tmp in list(zip(Y[:-1],
Y[1:]))])
f_k[3] = np.sum([1 if tmp[0] == 2 and tmp[1] == 1 else 0 for tmp in list(zip(Y[:-1],
Y[1:]))])
f_k[4] = np.sum([1 if tmp[0] == 2 and tmp[1] == 2 else 0 for tmp in list(zip(Y[:-1],
Y[1:]))])
# state feature
# i
f_k[5] = np.sum([1 if tmp == 1 else 0 for tmp in [Y[0]]])
f_k[6] = np.sum([1 if tmp == 2 else 0 for tmp in Y[:2]])
f_k[7] = np.sum([1 if tmp == 1 else 0 for tmp in Y[1:]])
f_k[8] = np.sum([1 if tmp == 2 else 0 for tmp in [Y[2]]])

# 生成全局特征向量
proba = np.sum(w_k*f_k)
# w的维度和f_k的维度匹配，一一对应
```

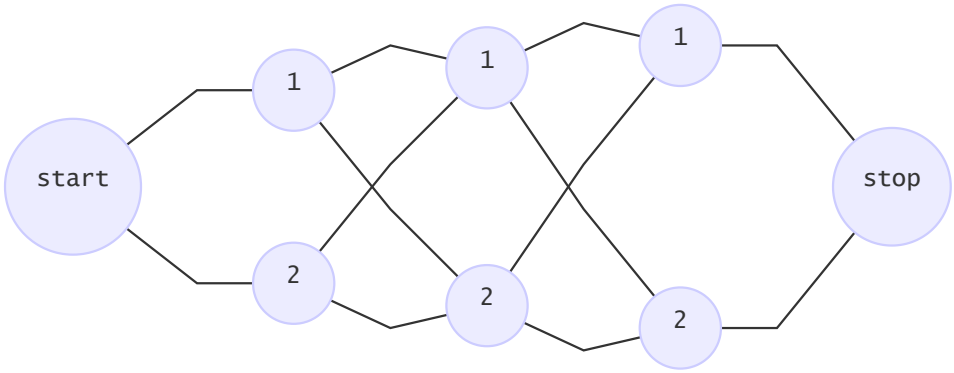
引用一下书中的解，注意看

$$
P(y|x) \propto \exp\left[\sum_{k=1}^{5}\lambda_k \sum_{i=2}^{3} t_k(y_{i-1}, y_i, x, i) + \sum_{k=1}^{4}\mu_k \sum_{i=1}^{3} s_k(y_i, x, i)\right]
$$

注意，按照这里红色部分的表达$\sum_{i=2}^{3}\sum_{i=1}^{3}$，实际上特征函数会遍历每一个可能的点和边。书中有这样一句**取值为 0 的条件省略**, 这个仔细体会下

例11.2

重复下题目， 其实就是做了符号说明



线性链条件随机场结构如上图

观测序列x，状态序列$y, i=1,2,3, n=3$，标记$y_i \in \{1,2\}$，假设$y_0 = start = 1, y_4 = stop = 1$，各个位置的随机矩阵为

$$M_1(x) = \begin{bmatrix} a_{01} & a_{02} \\ 0 & 0 \end{bmatrix}, M_2(x) = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$
$$M_3(x) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, M_4(x) = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

由M_i的定义

$$M_i(x) = \left[\exp\left(\sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i | x) \right) \right], i = 1, 2, \ldots, n+1$$

以及$y_i \in \{1,2\}$

可以知道，每个M_i中f_k对应的y_{i-1}, y_i都有两种取值，对应的组合就有四种

$$M(x) = \begin{bmatrix} \exp \sum_{k=1}^{K} w_k f_k(y[0], y[0]), \exp \sum_{k=1}^{K} w_k f_k(y[0], y[1]) \\ \exp \sum_{k=1}^{K} w_k f_k(y[1], y[0]), \exp \sum_{k=1}^{K} w_k f_k(y[1], y[1]) \end{bmatrix}$$

对应的红色部分组合使得M成为一个矩阵

这里注意M_1, M_4，理解这里的$y[0], y[1]$表示的是y的取值

$$M_1 \rightarrow y[0] \rightarrow start$$
$$M_4 \rightarrow y[4] \rightarrow end$$

这里重新整理一下

对于$y_0 = start = 1$

$$M(x) = \begin{bmatrix} \exp \sum_{k=1}^{K} w_k f_k(y[0], y[0]), \exp \sum_{k=1}^{K} w_k f_k(y[0], y[1]) \\ \exp \sum_{k=1}^{K} w_k f_k(y[1], y[0]), \exp \sum_{k=1}^{K} w_k f_k(y[1], y[1]) \end{bmatrix}$$

对于$y_4 = end = 1$

$$M(x) = \begin{bmatrix} \exp \sum_{k=1}^{K} w_k f_k(y[0], y[0]), \exp \sum_{k=1}^{K} w_k f_k(y[0], y[1]) \\ \exp \sum_{k=1}^{K} w_k f_k(y[1], y[0]), \exp \sum_{k=1}^{K} w_k f_k(y[1], y[1]) \end{bmatrix}$$

以上，取不到的值为0。

这里使用SymPy推导一下这个例子

```
from sympy import *
a01,a02, b11, b12, b21, b22, c11, c12, c21, c22  = symbols("a01, a02, \
                                                           b11, b12,b21, b22, \
                                                           c11, c12, c21, c22")
M1 = Matrix([[a01, a02],
             [0,   0]])
M2 = Matrix([[b11, b12],
             [b21, b22]])

M3 = Matrix([[c11, c12],
             [c21, c22]])

M4 = Matrix([[1, 0],
             [1, 0]])
Z = expand(M1*M2*M3*M4)
P = str(expand(M1*M2*M3*M4)[0]).replace(" ","").split("+")
# 体会各个路径之间关系
for i in range(2):
   for j in range(2):
       for k in range(2):
           logger.info(str(M1[0, i] * M2[i, j] * M3[j, k]))
print(Z)
print(P)
```

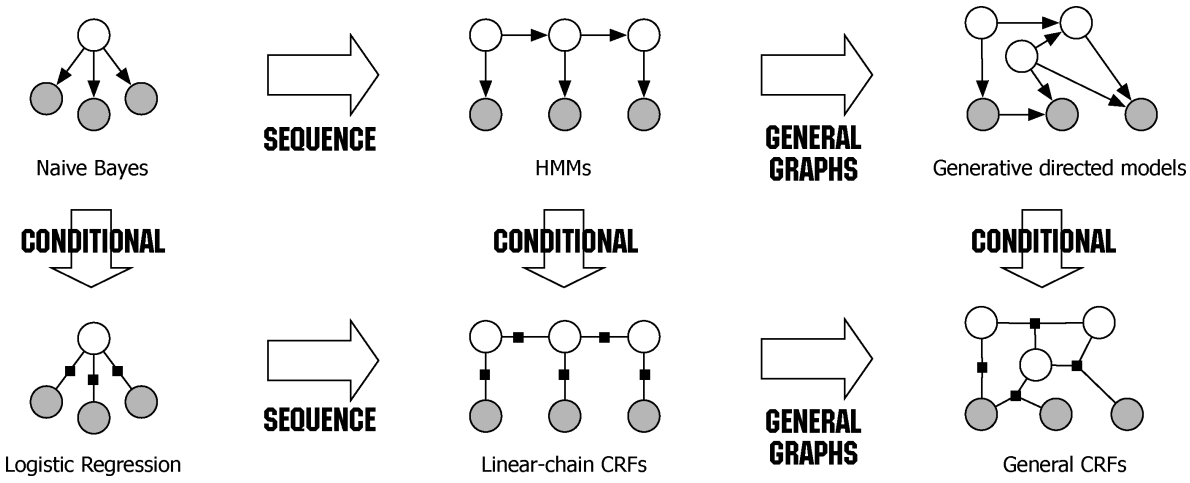
本章代码有设计这个例子的测试案例，可以参考。
这里有个点要注意下，书中强调了Z的**第一行和第一列**

$$Z(x) = \alpha_n^T(x) \cdot 1 = 1^T \cdot \beta_1(x)$$

例11.3

CRF与LR比较

都是对数线性模型



引用个图[5]

来自Sutton, Charles, and Andrew McCallum. "An introduction to conditional random fields." Machine Learning 4.4 (2011): 267-373.

上面一行是生成模型，下面一行是对应的判别模型。

应用

最后这两章的HMM和CRF真的是NLP方面有深入应用。HanLP的代码中有很多具体的实现。

从HMM推导出CRF

习题

EX11.1

图11.3无向图描述的概率图模型的因子分解式

$$P(Y)=\frac{1}{Z}\Psi_{C1}(Y_{C1})\Psi_{C2}(Y_{C2})$$

EX11.3

11.3 写出条件随机场模型学习的梯度下降算法

参考

1.
2.

↑ top

CH12 统计学习方法总结

前言

章节目录

这一章的内容很简洁, 但是信息量很大.

1. 适用问题
2. 模型
3. 学习策略
4. 学习算法

导读

- 这一章首先用一个表概括了十个**方法**，注意，这里是十个方法，而不是十个模型，书中脚注了其中EM算法比较特殊，是一个一般的方法，没有具体模型。
  其余方法会对应生成模型或者判别模型。
- 监督学习包括**分类, 标注**和回归。

- 这一章有说明这本书章节划分，前半部分分类，后面两章标注，中间EM算法可以用于生成模型的非监督学习。
- 本书中生成模型就两个，NB用于分类，HMM用于标注。
- 关于概率模型与非概率模型，体会一下细节，尤其是既可以看作概率模型，又可看作非概率模型的情况。
-

统计学习方法

方法=模型+策略+算法

监督学习, 非监督学习, 强化学习都有这样的三要素.

这里回顾一下第一章的统计学习三要素:

1. 模型
    1. 监督学习中, 模型就是所要学习的条件概率分布或者决策函数.
2. 策略
    1. 统计学习的目标在于从假设空间中选取最优模型.
    2. 损失函数度量一次预测的好坏; **风险函数**度量平均意义下模型预测的好坏.
    3. 经验风险最小化(ERM)与结构风险最小化(SRM)
    4. 经验风险或者结构风险是最优化的目标函数.
3. 算法
    1. 统计学习基于训练数据集, 根据学习策略, 从假设空间中选择最优模型, 最后需要考虑用干什么样的计算方法求解**最优模型**.
    2. 统计学习问题转化为最优化问题.
        1. 有显式解析解, 对应的最优化问题比较简单
        2. 通常解析解不存在, 需要通过数值计算的方式求解.
    3. 算法需要解决的问题是如何找到**全局最优解**, 并且求解的过程非常高效.

不同视角

这本书的内容可以从多个角度进行划分

1. 简单分类方法
    1. 感知机
    2. k近邻法
    3. 朴素贝叶斯法
    4. 决策树
2. 复杂分类方法
    1. 逻辑斯谛回归模型
    2. 最大熵
    3. 支持向量机
    4. 提升方法
3. 标注方法
    1. 隐马尔科夫模型
    2. 条件随机场

模型

分类问题与标注问题都可以认为是从输入空间到输出空间的映射。

他们可以写成条件概率分布$P(Y|X)$或者决策函数$Y=f(x)$的形式。

概率模型和非概率模型

对应**概率模型**和**非概率模型**.

1. 概率模型(由条件概率表示的模型)
    1. 朴素贝叶斯
    2. 隐马尔科夫模型
2. 非概率模型(由决策函数表示的模型)
    1. 感知机
    2. k近邻
    3. 支持向量机
    4. 提升方法
3. 概率模型和非概率模型
    1. 决策树
    2. 逻辑斯谛回归模型
    3. 最大熵模型
    4. 条件随机场

生成模型和判别模型

1. 判别模型
    1. 直接学习条件概率分布$P(Y|X)$或者决策函数$Y=f(X)$的方法为判别方法，对应的模型为判别模型。
    2. 感知机，k近邻，决策树，逻辑斯谛回归模型，最大熵模型，支持向量机，提升方法，条件随机场
2. 生成模型
    1. 先学习联合概率分布$P(X,Y)$，从而求得条件概率分布$P(Y|X)$的方法是生成方法，对应的模型是生成模型。
    2. 朴素贝叶斯，隐马尔科夫模型

线性模型和非线性模型

1. 线性模型
    1. 感知机
2. 对数线性模型
    1. 逻辑斯谛回归模型
    2. 最大熵模型
    3. 条件随机场
3. 非线性模型
    1. k近邻
    2. 决策树
    3. 支持向量机(核函数)
    4. 提升方法

生成与判别, 分类与标注

这部分书中图12.1，可以参考CH11中引用的图来进一步理解。

	判别	**生成**
分类	LR, ME	NB
标注	CRF	HMM

这个对应关系可以通过后面的概括总结表格进一步加深，LR和CRF对应， 其模型类型，学习策略，损失函数以及学习算法都是一样的，只是解决的问题有差异，LR解决分类问题， CRF解决标注问题。

NB和HMM也是一对(GD Pair)，他们在模型类型，学习策略，学习算法都是一样的，注意这里损失函数写的有一定的差异，CRF的损失函数是对数似然损失，实际上书中CH06中的描述也是对数似然损失，在本章描述的逻辑斯谛损失是从间隔角度考虑的另外视角。

学习策略

损失函数

注意，书中这里描述的是，**在二分类的监督问题中**，后面会在这个基础上做推广。

1. 合页损失

   线性支持向量机

   $max(0, 1 - yf(x))$

2. 逻辑斯谛损失函数

   逻辑斯谛回归模型与最大熵模型

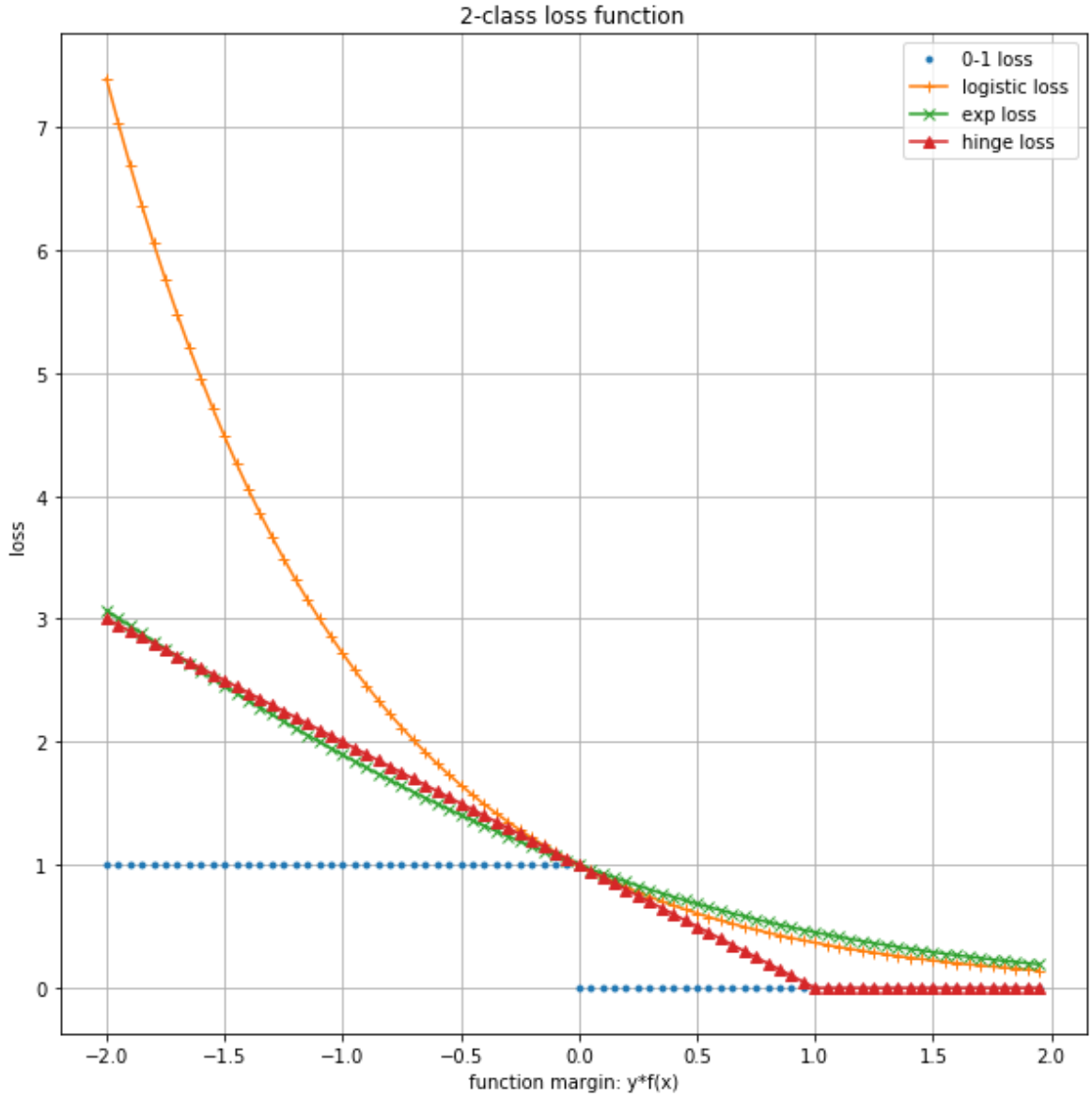
   $log(1 + exp(-yf(x)))$

3. 指数损失函数

   提升方法

   $exp(-yf(x))$

三种损失函数都是0-1损失函数的上界.



上面这个图有几点要注意:

1. logistic loss，里面的对数是2
2. 另外，这些函数在0右侧的部分，都是有值的。
3. 分类问题的损失，实现二分类任务

这几个模型，用在分类问题上，可以有一种统一的表达来描述损失函数。这会引入**经验风险最小化**和**结构风险最小化**。

学习的策略是优化以下结构风险函数

$$\min_{f\in H}\frac{1}{N}\sum_{i=1}^{N}L(y_i,f(x_i))+\lambda J(f)$$

第一项为经验风险(经验损失)，第二项为正则化项

正则化

提升方法没有显式的正则化项，可以通过early stop控制停止

二分类推广

1. 推广到多分类
2. **标注问题的条件随机场可以看成是分类问题的最大熵模型的推广**，这个通过图模型来理解更好理解一点
3. 概率模型的学习可以形式化为极大似然估计或贝叶斯估计的极大后验概率估计
4. 决策树CH05的学习策略是正则化的极大似然估计，损失函数是对数损失函数，正则化项是决策树的复杂度。
5. 逻辑斯谛回归模型与最大熵模型CH06，条件随机场的学习策略既可以看成是极大似然估计，又可以看成是极小化逻辑斯谛损失。 书中讲的是极大似然估计，这样方便对比LR和CRF。
6. 朴素贝叶斯模型CH04，隐马尔科夫模型CH10的非监督学习也是极大似然估计或极大后验概率估计，但这时模型含有隐变量。

学习算法

1. 朴素贝叶斯法CH04和隐马尔科夫模型CH10
2. 感知机CH02，逻辑斯谛回归模型CH06，最大熵模型CH06，条件随机场CH11
3. 支持向量机CH07
4. 决策树CH05
5. 提升方法CH08
6. EM算法CH09
7. NB和HMM的监督学习，最优解就是极大似然估计值，可以由概率计算公式直接计算。之前看NB其实就是计数查表，这种要有大的语料库进行统计，所谓学的多，就知道的多。

方法	适用问题	模型特点	模型类型	学习策略	学习的损失函数	学习算法
Peceptron	二类分类	分离超平面	判别模型	极小化误分点到超平面距离	误分点到超平面距离	SGD
KNN	多类分类, 回归	特征空间, 样本点	判别模型			
NB	多类分类		生成模型	MLE, MAP	对数似然损失	概率计算公式, EM算法
DT	二类分类		判别模型	正则化的极大似然估计	对数似然损失	特征选择, 生成, 剪枝
LR Maxent	多类分类		判别模型			
SVM	二类分类		判别模型			
AdaBoost	二类分类		判别模型			
EM	概率模型参数估计	含隐变量的概率模型				
HMM	标注	观测序列与状态序列的联合概率分布模型	生成模型			
CRF	标注		判别模型			

TODO：更新上面的表

特征空间

↑ top

CH13 无监督学习概论

前言

章节目录

导读

- 在这部分强调了**样本(实例)**，由特征向量组成。
- 无监督学习的基本问题是聚类，降维，概率估计，对应的输出是类别，转换，概率。
- 无监督学习的模型是$z=g_\theta(x)$(硬聚类)，条件概率分布$P_\theta(z|x)$（软聚类）**或条件概率分布**$P_\theta(x|z)$（概率模型估计）
- 软聚类可以看成是概率模型估计问题，根据贝叶斯公式

$$P_\theta(z|x)=\frac{P_\theta(x|z)P_\theta(z)}{P_\theta(x)}\propto \underbrace{P_\theta(z)}_{\text{假设服从均匀分布}} \quad P_\theta(x|z)\propto P_\theta(x|z)$$

  假设先验概率服从均匀分布，$P_\theta(z)$就是常数，先验后验成正比。
- 训练数据可以用$M\times N$矩阵表示，注意这里矩阵的每一行对应特征，每一列对应一个样本，回想在之前监督学习部分，也是用N表示样本的个数。但是这里面其实稍微有一点点尴尬，在监督学习部分，特征的维数用n表示，样本数量用N表示，在无监督学习部分，特征的维数用m表示，样本的数量用n表示。
- 无监督学习可以用于数据分析或者监督学习的前处理
- `无监督学习通常需要大量的数据，因为对数据隐藏的规律的发现需要足够的观测`。反过来看，当我们拥有大量数据的时候，可以考虑通过无监督学习的方式来发现数据中隐藏的规律。有的时候我们需要从问题的角度出发来寻找解决方案，而有的时候，我们需要从自身的角度出发，来看能做些什么。不同的岗位看问题的角度不同，但是岗位的差异并不应该限制你思考问题的维度。
- 这章的参考文献和第一章的基本一样，去掉了Sutton的强化学习。
- 降维部分有提到manifold，这个书中应该是没有展开，可以参考scikit-learn中相关部分。

无监督学习基本原理

符号说明

训练数据集X

$$X=\begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & & \vdots \\ x_{M1} & \cdots & x_{MN} \end{bmatrix}$$

基本问题

聚类

发现数据中的**纵向结构**

硬聚类

$$z = g_\theta(x)$$

软聚类

$$P_\theta(z|x)$$

$x \in X$是样本的向量

$z \in Z$是样本的类别

θ是参数

降维

发现数据中的**横向结构**

$$z = g_\theta(x)$$

$x \in X$ 是样本的高维向量

$z \in Z$ 是样本的低维向量

θ 是参数

g 可以是线性函数，也可以是非线性函数。

概率模型估计

$$P_\theta(x|z)$$

找到最有可能生成数据的结构和参数。

机器学习三要素

模型，策略，算法，这部分概括下和监督学习主要的区别在模型差异，至于策略和算法，大体路子和监督学习差不多，具体问题具体分析。

关于模型，上一小节有介绍，这一小节简单重复下，将在下一小节展开，并在后续章节详细说明。

关于策略，各自目标函数的优化过程，比如**聚类**样本与所属类别中心距离的最小化，**降维**过程信息损失的最小化，**概率模型估计**过程中生成数据概率的最大化。

关于算法，迭代算法，梯度下降打天下。

无监督学习方法

聚类

CH14，主要是数据硬聚类，Kmeans，层次聚类，软聚类在之前的EM算法部分已经讲过，比如高斯混合模型。

降维

CH15，CH16，主成分分析，以及奇异值分解。

这部分先写了第16章介绍PCA，然后说的第15章。因为PCA是无监督学习方法，SVD是基础学习方法。后面话题分析部分也是这样的顺序。

话题分析

LSA，PLSA，LDA是无监督学习方法，MCMC是基础的学习方法。

图分析

PageRank，Page是个人名。

发现隐藏在图中的统计规律或潜在结构。

参考

CH14 聚类方法

前言

章节目录

1. 聚类的基本概念
    1. 相似度或距离
    2. 类或簇
    3. 类与类之间的距离
2. 层次聚类
3. k均值聚类
    1. 模型
    2. 策略
    3. 算法
    4. 算法特性

导读

- Kmeans是1967年由MacQueen提出的，注意KNN也是1967年提出的，作者是Cover和Hart。

聚类的基本概念

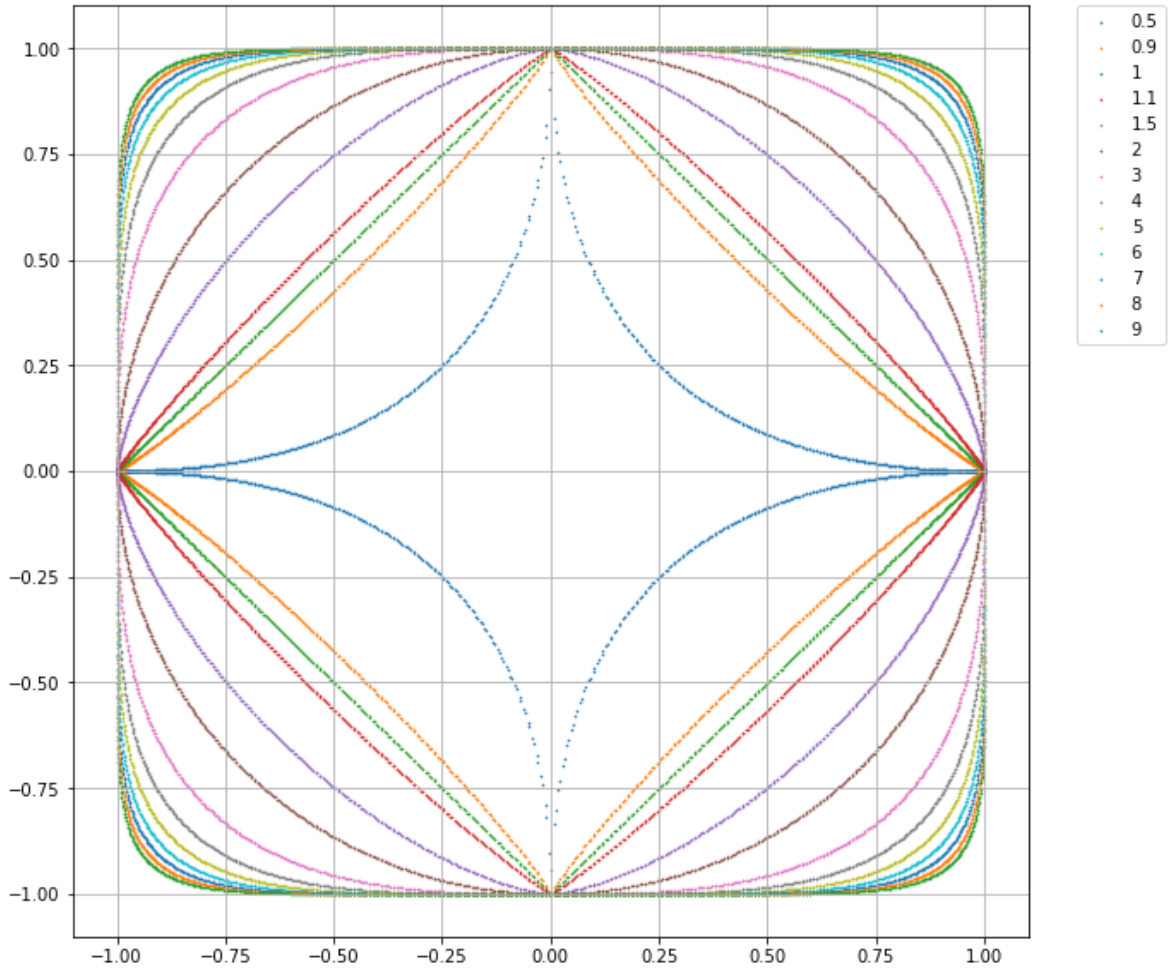
以下实际上是算法实现过程中的一些属性。

矩阵X表示样本集合，$X \in \mathbf{R}^m, x_i, x_j \in X, x_i = (x_{1i}, x_{2i}, \dots, x_{mi})^{\mathrm{T}}, x_j = (x_{1j}, x_{2j}, \dots, x_{mj})^{\mathrm{T}}$，$n$个样本，每个样本是包含$m$个属性的特征向量，

距离或者相似度

闵可夫斯基距离

$$d_{ij} = \left(\sum_{k=1}^{m} |x_{ki} - x_{kj}|^p \right)^{\frac{1}{p}}$$
$$p \geq 1$$



这个图可以再展开

马哈拉诺比斯距离

马氏距离

$$d_{ij} = \left[(x_i - x_j)^{\mathrm{T}} S^{-1} (x_i - x_j)\right]^{\frac{1}{2}}$$

相关系数

$$r_{ij} = \frac{\sum_{k=1}^{m}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\left[\sum_{k=1}^{m}(x_{ki} - \bar{x}_i)^2 \sum_{k=1}^{m}(x_{kj} - \bar{x}_j)^2\right]^{\frac{1}{2}}}$$

$$\bar{x}_i = \frac{1}{m}\sum_{k=1}^{m} x_{ki}$$

$$\bar{x}_j = \frac{1}{m}\sum_{k=1}^{m} x_{kj}$$

夹角余弦

$$s_{ij} = \frac{\sum_{k=1}^{m} x_{ki} x_{kj}}{\left[\sum_{k=1}^{m} x_{ki}^2 \sum_{k=1}^{m} x_{kj}^2\right]^{\frac{1}{2}}}$$

距离和相关系数的关系

其实书上的这个图，并看不出来距离和相关系数的关系，但是书中标注了角度的符号。

类或簇

类与类之间的距离

类和类之间的距离叫做linkage，这些实际上是算法实现过程中的一些属性。

类的特征包括：均值，直径，样本散布矩阵，样本协方差矩阵

类与类之间的距离：最短距离，最长距离，中心距离，平均距离。

类G_p和类G_q

最短(single linkage)

$D_{pq} = \min\{d_{ij}|x_i \in G_p, x_j \in G_q\}$

最长(complete linkage)

$D_{pq} = \max\{d_{ij}|x_i \in G_p, x_j \in G_q\}$

平均(average linkage)

$D_{pq} = \frac{1}{n_p n_q} \sum_{x_i \in G_p} \sum_{x_j \in G_q} d_{ij}$

中心

$D_{pq} = d_{\bar{x}_p \bar{x}_q}$

层次聚类

层次聚类**假设**类别之间存在层次结构。

层次聚类可以分成聚合聚类和分裂聚类。

聚合聚类三要素：

1. 距离或相似度：闵可夫斯基，马哈拉诺比斯，相关系数，余弦距离
2. 合并规则：类间距离最小，最短，最长，中心，平均
3. 停止条件：类的个数达到阈值，类的直径达到阈值

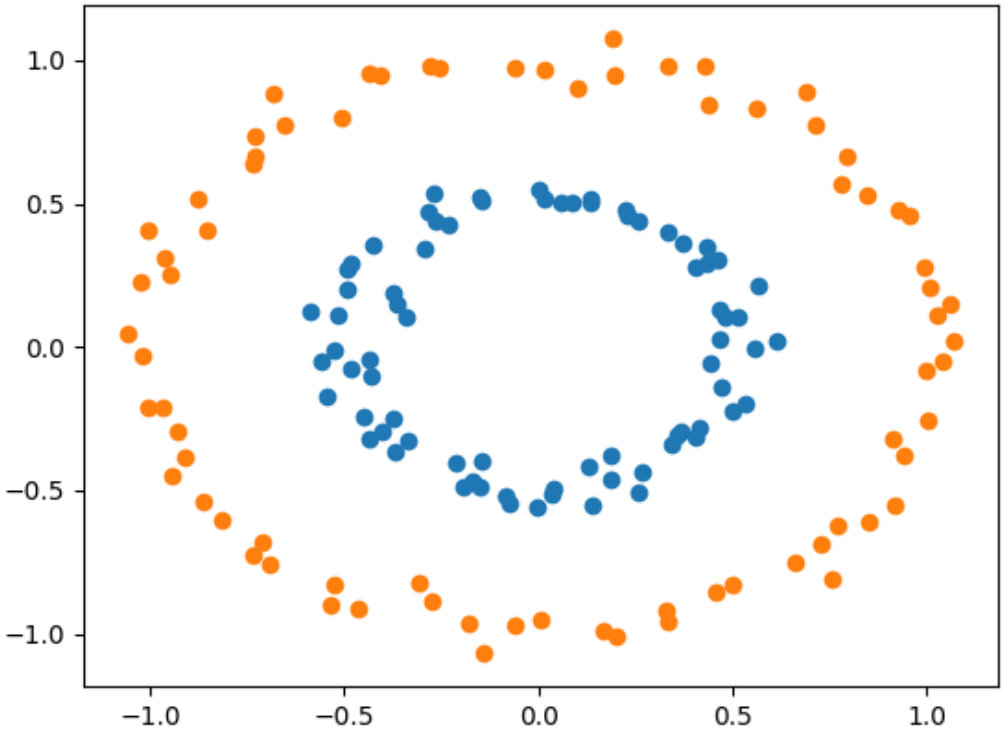
算法14.1

输入：n个样本组成的集合X

输出：对样本的一个层次化聚类C

1. 计算n个样本两两之间的欧氏距离$\{d_{ij}\}$，记作矩阵$D = [d_{ij}]_{n \times n}$
2. 构造n个类，每个类只包含一个样本
3. 合并类间距离最小的两个类，其中最短距离为类间距离，构建一个新类。
4. 计算新类与当前各类之间的距离。如果类的个数是1， 终止计算，否则回到步骤3。

这个算法复杂度比较高$O(n^3 m)$



如图，采用层次聚类实现circle的划分。

Kmeans聚类

注意对于kmeans来说，距离采用的是欧氏距离**平方**，这个是个特点。

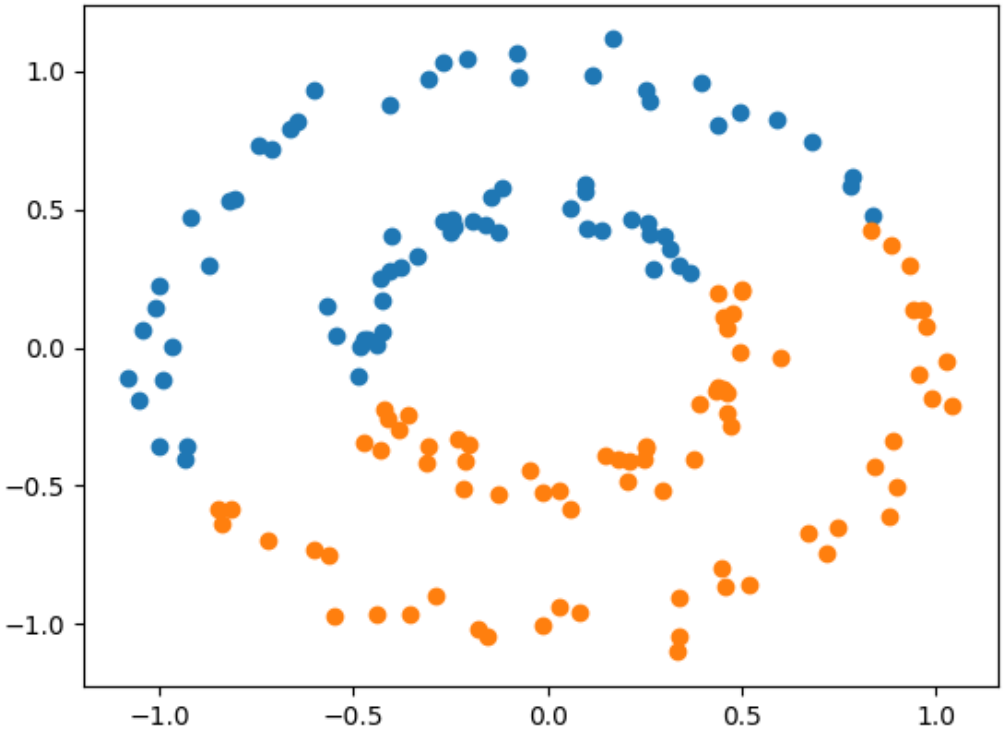
算法14.2

输入：n个样本的集合X

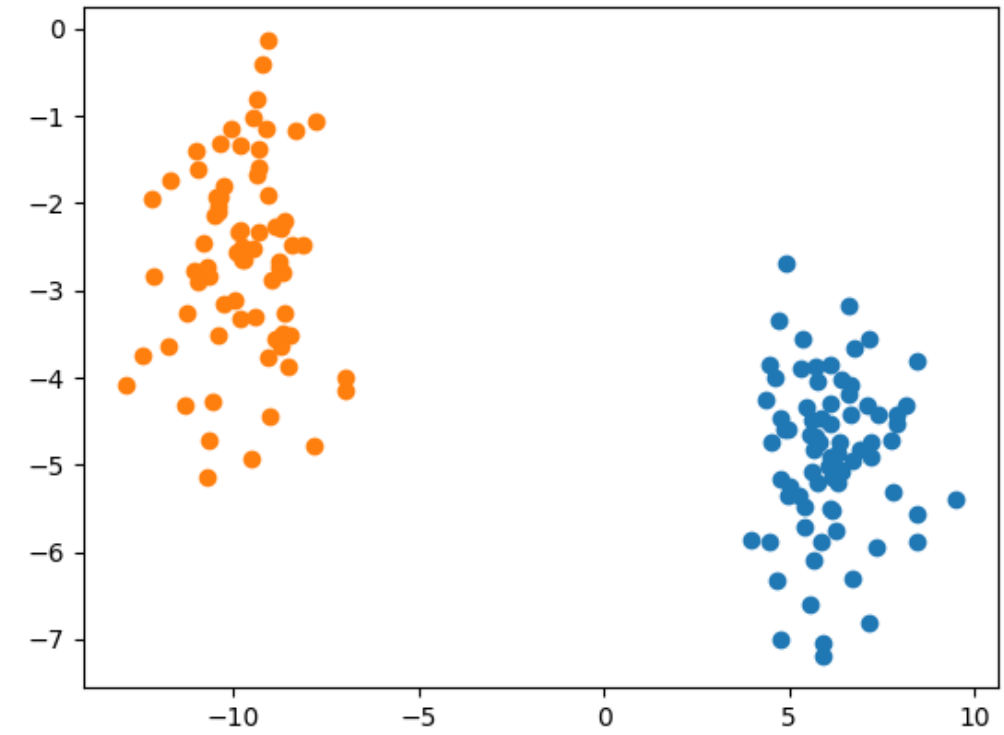
输出：样本集合的聚类C^*

1. 初始化。
2. 对样本进行聚类。
3. 计算类的中心。
4. 如果迭代收敛或符合停止条件，输出$C^* = C^{(t)}$

对于Cirlce数据，如果采用kmeans聚类，得到结果如下



Blob数据采用kmeans结果如下



例子

14.1

这个例子里面，直接给定的是距离矩阵，类间距离选择的是最小距离。

14.2

这个例子很有意思，实际上，最好的划分，不一定是书中给的答案的划分。这也说明了初值的选择，对于kmeans算法最后的结果影响比较重要。

在后面的初始类选择部分，对此做了解释。但是实际上在做到这个例子的时候应该就能想到这个问题，书中选择的数据很典型。

参考

CH15 奇异值分解

前言

章节目录

1. 奇异值分解的定义与性质
    1. 定义与定理
    2. 紧奇异值分解与截断奇异值分解
    3. 几何解释
    4. 主要性质
2. 奇异值分解的计算
3. 奇异值分解与矩阵近似
    1. 弗罗贝尼乌斯范数
    2. 矩阵的最优近似
    3. 矩阵的外积展开式

导读

- SVD是线性代数的概念，但在统计学中有广泛应用，PCA和LSA中都有应用，在本书中定义为基础学习方法。
- SVD是矩阵分解方法，特点是分解的矩阵正交。还有另外一种矩阵分解方法叫做NMF，其特点是分解的矩阵非负。
- 奇异值分解是在平方损失意义下对矩阵的最优近似，即**数据压缩**。图像存储是矩阵，那么图像也可以用SVD实现压缩。
- 任意给定一个实矩阵，其奇异值分解一定存在，但并不唯一。Σ是唯一的，U和V^{T}是可变的。
- 奇异值分解有明确的几何意义，事实上，整个线性代数都有明确的几何意义。
- 提到旋转或**反射变换**。关于反射变换，定点或者定直线对称，定点的叫做中心反射，定直线的叫做轴反射。
- 奇异值分解可以扩展到Tensor。
- 推荐阅读部分推荐了MIT的18.06SC，这里也推荐下3Blue1Brown，快速建立线性代数相关定义的几何直观，如果有具体的哪个点不清楚，不形象，也可以考虑查下。其实，还得多用。
- 接上条，MIT 18.06SC的教材，《Introduction to linear algebra》这个书不错，推荐看。
- 图15.1好好体会下，那个图里面，左上角的图形，两个轴是基矢量，单位长度，所以右侧的图基矢量的长度是σ_1, σ_2，这说明了奇异值的意义。另外，思考下怎么实现。
- $Ax = U\Sigma V^{\mathrm{T}}x$，所以，按照$V^{\mathrm{T}}, \Sigma, U$的顺序乘
- W有时候表示能量，在这章$W = A^{\mathrm{T}}A$

线性代数回顾

这部分内容主要参考3Blue1Brown中Essence of Linear Algebra，目录列举如下：

- Chapter 1: Vectors, what even are they?

- Chapter 2: Linear combinations, span and bases
- Chapter 3: Matrices as linear transformations
- Chapter 4: Matrix multiplication as composition
- Chapter 5: The determinant
- Chapter 6: Inverse matrices, column space and null space
- Chapter 7: Dot products and cross products
- Chapter 8: Cross products vis transformations
- Chapter 9: Change of basis
- Chapter 10: Eigenvectors and eigenvalues
- Chapter 11: Abstract vector spaces

向量

在计算机里面，向量就是一个有序的列表。

x轴和y轴的交点是原点，是整个空间的中心和所有向量的根源。向量中的数字代表从原点出发依次在每个轴上走多远，最后可以到达向量的终点。

为了把**向量和点分开**，向量通常竖着写，用方括号包围$\begin{bmatrix}1\\2\end{bmatrix}$，而点用$(1,2)$表示。

线性代数的每个主题都围绕着向量加法和向量数乘。

向量的加法定义是唯一一个允许向量离开原点的情形。

在现在理论中，向量的形式并不重要， 箭头，一组数，函数等都可以是向量。只要向量相加和数乘的概念遵守以下规则即可，这些规则叫做公理：

$$
\begin{aligned}
&\vec{u}+(\vec{v}+\vec{w})=(\vec{u}+\vec{v})+\vec{w}\\
&\vec{v}+\vec{w}=\vec{w}+\vec{v}\\
&There\ is\ a\ vector\ 0\ such\ that\ 0+\vec{v}=\vec{v}\ for\ all\ \vec{v}\\
&For\ every\ vector\ \vec{v}\ there\ is\ a\ vector\ -\vec{v}\ so\ that\ \vec{v}+(-\vec{v})=0\\
&a(b\vec{v})=(ab)\vec{v}\\
&1\vec{v}=\vec{v}\\
&a(\vec{v}+\vec{w})=a\vec{v}+a\vec{w}\\
&(a+b)\vec{v}=a\vec{v}+b\vec{v}
\end{aligned}
$$

向量加法

$$
\begin{bmatrix}x_1\\y_1\end{bmatrix}+\begin{bmatrix}x_2\\y_2\end{bmatrix}=\begin{bmatrix}x_1+x_2\\y_1+y_2\end{bmatrix}
$$

向量数乘

Scaling，缩放的过程。用于缩放的数字，叫做标量，Scalar。

在线性代数中，数字的作用就是缩放向量。

$$
2\cdot\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}2x\\2y\end{bmatrix}
$$

基

向量可以考虑成是把**基**缩放并且相加，当我们用一组数字描述向量时，他们都依赖于我们正在使用的基。

向量的线性组合与空间张成。

我们通常用向量的终点代表向量，起点位于原点。

线性变换

1. 直线仍然变成直线
2. 原点保持不变

保持网格平行并等距的变换，向量作为输入输出。

一个二维线性变换，仅由四个数字完全确定。$\begin{bmatrix} i_1 & j_1 \\ i_2 & j_2 \end{bmatrix}$描述了线性变换。

$$\begin{bmatrix} i_1 & j_1 \\ i_2 & j_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x\underbrace{\begin{bmatrix} i_1 \\ j_1 \end{bmatrix}}_{basis} + y\underbrace{\begin{bmatrix} i_2 \\ j_2 \end{bmatrix}}_{basis} = \begin{bmatrix} xi_1 + yi_1 \\ xi_2 + yj_2 \end{bmatrix}$$

逆时针旋转90度(90 rotation counterclockwise)的线性变换矩阵，可以从x-y的基旋转之后的值来得到。

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Shear变换

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

线性变换是操纵空间的一种手段，他保持网格平行等距分布，且原点保持不动。

矩阵乘法

矩阵乘法的几何意义是一个线性变换之后再跟一个线性变换，两个线性变换的相继作用。

顺序从右到左，因为我们函数在变量左侧。

这部分提到`Good Explanation > Symbolic proof`

二维平面的结果，可以完美的推广到三维的空间。三维线性变换由基向量的去向完全决定。

行列式

`The purpose of computation is insight, not numbers.-Richard Hamming`

1. 一个矩阵的行列式的绝对值为k说明将原来一个区域的面积变为k倍，变成0了说明降维了，平面压缩成了线，或者点。

   行列式为0说明降维了。

2. 行列式可以为负数，说明翻转了。这是二维空间的定向，三维空间的定向是"右手定则"

$$\det(\begin{bmatrix} a & c \\ b & d \end{bmatrix}) = ad - bc$$

线性方程组

Linear system of equations

$A\vec{x} = \vec{v}$

寻找一个向量$\vec{x}$，经过线性变换后和$\vec{v}$重合

A^{-1}的核心性质就是

$$A^{-1}A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

这个也叫恒等变换。

$$A^{-1}A\vec{x} = \vec{x} = A^{-1}\vec{v}$$

$\det(A) \neq 0 \Rightarrow A^{-1}exists$

秩

Rank代表变换后空间的维数。

列空间

矩阵的列告诉我们基向量变换之后的位置，列空间就是矩阵的列所张成的空间。

这部分实际上是书中附录D介绍的内容。

秩的定义是列空间的维数。

满秩，就是秩等于列数

零向量一定在列空间内，满秩变换中，唯一能落在原点的就是零向量自身。

零空间

变换后，落在零向量的点的集合是零空间，或者叫核，所有可能解的集合。

非方阵

3×2，矩阵是把二维空间映射到三维空间上，因为矩阵有两列，说明输入空间有两个基向量，三行表示每一个基向量在变换后用三个独立的坐标来描述。

2×3，矩阵是把三维空间映射到二维空间上，因为矩阵有三列，说明输入空间有三个基向量，二行表示每一个基向量在变换后用二个独立的坐标来描述。

1×2，矩阵是把二维空间映射到一维数轴上，因为矩阵有两列，说明输入空间有两个基向量，一行表示每一个基向量在变换后用一个独立的坐标来描述。

叉乘

是线性的，一旦知道是线性的，就可以引入对偶性的思考了。

点乘与叉乘非常重要。

转移矩阵

$$A^{-1}MA$$

暗示着数学上的转移作用，中间的矩阵代表所见到的变换，外侧两个矩阵代表着转移作用，也就是视角上的转换。矩阵乘积仍然代表同一个变换M，只不过是其他人的视角。

特征向量与特征值

特征向量，在变换过程中留在了自己张成的空间内，这样的向量叫做特征向量。在变换过程中只受到拉伸或者压缩。

特征向量在变换中拉伸或者压缩的比例因子叫做特征值。

理解线性变换的作用的关键往往较少依赖于你的特定坐标系。

$$\begin{aligned} A\vec{v} &= (\lambda I)\vec{v} \\ A\vec{v} - (\lambda I)\vec{v} &= 0 \\ (A - \lambda I)\vec{v} &= 0 \\ \det(A - \lambda I) &= 0 \end{aligned}$$

这部分，行列式为0的几何意义很重要。如果没有实数解，说明没有特征向量。

空间

Determinant and eigenvectors don't care about the coordinate system.
行列式告诉你一个变换对面积的缩放比例，特征向量则是在变换中保留在他所张成的空间中的向量，这两者都是暗含与空间中的性质，坐标系的选择并不会改变他们最根本的值。

函数实际上只是另一种向量。

- 函数的线性变换，比如微积分中的导数，有时候会用**算子**来表示**变换**的意思。
- 求导具有可加性和成比例性。
- 函数空间趋近于无限维
- 多项式空间，求导
- 矩阵向量乘法和矩阵求导看起来是不相关的，但实际上是一家人。

线性代数	**函数**
线性变换	线性操作
点乘	内积
特征向量	特征函数

只要处理的对象有合理的数乘和相加的概念，只要定义满足公理，就能应用线性代数中的结论。

抽象性带来的好处是我们能得到一般性的结论。

最后补充一点，关于视频中描述一个变换的中间步骤的过程，可以参考Chapter 14的7:45左右的视频内容体会。

虽然，前面的内容很精彩，但是看完之后，再看书中内容，可能依然....懵...吧，没事，继续看就好。

奇异值分解定义与性质

定义

矩阵的奇异值分解是指将$m \times n$实矩阵A表示为以下三个实矩阵乘积形式的运算

$$A = U\Sigma V^{\mathrm{T}}$$

中间有一句，`可以假设正交矩阵$V$的列的排列使得对应的特征值形成降序排列`。这句怎么理解？

列是轴，实际上不同列的排列，对应的是坐标轴的顺序，不同坐标系顺序的选择，和实际上拿到的最后的向量是没有关系的。

完全奇异值分解：$A = U\Sigma V^{\mathrm{T}}$

紧奇异值分解：$A = U_r\Sigma_r V_r^{\mathrm{T}}$

截断奇异值分解：$A = U_k\Sigma_k V_k^{\mathrm{T}}$

几何解释

$A_{m\times n}$表示了一个从n维空间$\mathbf{R}^n$到m维空间$\mathbf{R}^m$的一个**线性变换**

$$\begin{gathered} T: x \to Ax \\ x \in \mathbf{R}^n \\ Ax \in \mathbf{R}^m \end{gathered}$$

线性变换可以分解为三个简单的变换：

1. 坐标系的旋转或反射变换，V^{T}
2. 坐标轴的缩放变换，Σ
3. 坐标系的旋转或反射变换，U

这里面注意，A其实就是**线性变换**，关于线性变换的概念，在前面有整理。

这里**比较重要的一个图**是图15.1。

主要性质

1. AA^{T}和$A^{\mathrm{T}}A$的特征分解存在，且可由矩阵A的奇异值分解的矩阵表示；
2. 奇异值，左奇异向量，右奇异向量之间的关系
3. 矩阵A的奇异值分解中，奇异值是唯一的，但是矩阵U和V不是唯一的，所以numpy.linalg.svd中有参数控制是否输出U和V
4. Rank
5. r

奇异值分解的计算

主要就是一个例子，15.5

这个例子在最后总结的时候，说明了SVD算法最重要的就是$A^{\mathrm{T}}A$的特征值的计算。特征值有了之后Σ就拿到了，所以在numpy里面，SVD可以选择输出Σ还是U,Σ,V^{T}。这里只有Σ是确定的，U和V^{T}是不唯一的。

有个定义叫做$W=A^{\mathrm{T}}A$，回顾一下聚类那章定义了W是能量，那么我们保存80%-90%的矩阵能量，大概能起到保留重要信息，滤除噪声的作用。

奇异值分解与矩阵近似

奇异值分解也是一种矩阵近似的方法，这个近似是在**弗罗贝尼斯范数**意义下的近似。

$$\|A\|_F=\left(\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}^2\right)^{\frac{1}{2}}$$
$$A\in\mathbf{R}^{m\times n},A=[a_{ij}]_{m\times n}$$

矩阵的弗罗贝尼斯范数是向量的L_2范数的直接推广，对应着机器学习里面的平方损失函数。矩阵范数(matrix norm)也是一个很大的概念，详细内容可以扩展下[5]。

$$A\in\mathbf{R}^{m\times n}$$
$$A=U\Sigma V^{\mathrm{T}}$$
$$\Sigma=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_{\mathrm{n}})$$
$$\|A\|_F=(\sigma_1^2+\sigma_2^2+\cdots+\sigma_{\mathrm{n}}^2)^{\frac{1}{2}}$$

矩阵的最优近似

$$\|A-X\|_F=\min_{S\in\mathcal{M}}\|A-S\|_F$$
$$\|A-X\|_F=(\sigma_{k+1}^2+\sigma_{k+2}^2+\cdots+\sigma_n^2)^{\frac{1}{2}}$$

矩阵的外积展开式

$$\begin{aligned}A&=\sigma_1u_1v_1^{\mathrm{T}}+\sigma_2u_2v_2^{\mathrm{T}}+\cdots+\sigma_nu_nv_n^{\mathrm{T}}\\&=\sum_{k=1}^{n}A_k\\&=\sum_{k=1}^{n}\sigma_ku_kv_k^{\mathrm{T}}\end{aligned}$$

其中，$u_kv_k^{\mathrm{T}}$为$m\times n$矩阵

例子

15.1

numpy中linalg提供了svd函数。

15.2

15.3

这两个习题主要是看紧奇异值分解和截断奇异值分解。

15.4

奇异值分解，就是一个矩阵A拆成3个矩阵，按照顺序把三个矩阵乘过去，会恢复成原来的矩阵A。

15.5

这个可以复习下正交矩阵，单位化，正交基， 标准正交基

15.6

这个例子看下v_1的定义，以及后面的v_1^{T}，体会一下维度的变化，$u_k v_k^{\mathrm{T}}$的维度是$m \times n$

习题

15.5

这个例子很有意思，实际上这个图，是一种稀疏表示，而矩阵是一种稠密的表示，但是这个矩阵也是一个稀疏矩阵。

还有，这里提到了二部图，实际上在这本书里面应该没讲过二部图。

	u_1	u_2	u_3	u_4	u_5
q_1	0	20	5	0	0
q_2	10	0	0	3	0
q_3	0	0	0	0	1
q_4	0	0	1	0	0

这个矩阵恢复出来就是这样的，SVD分解之后，U应该是quary的相似度

参考

CH16 主成分分析

前言

章节目录

1. 总体主成分分析
    1. 基本想法
    2. 定义和导出
    3. 主要性质
    4. 主成分的个数
    5. 规范化变量的总体主成分
2. 样本主成分分析

1. 样本主成分的定义和性质
2. 相关矩阵的特征值分解算法
3. 数据矩阵的奇异值分解算法

导读

- 这部分内容介绍了**总体**主成分分析的定义，定理与性质，并在第二节介绍了**样本**主成分分析，介绍了主成分分析的算法。
- PCA的基本想法是由少数不相关的变量来代替相关的变量，用来表示数据，并且要求能够保留数据中的大部分信息。注意这个不是特征选择，得到的主成分是线性无关的新变量。
- 所谓线性相关的x_1和x_2就是说知道x_1的值的情况下，x_2的预测不是完全随机的。
- 主成分分析的结果可以作为其他机器学习方法的输入。
- 参考文献4应该是这本书引用的日期最新的Journal Article了，2014年的，文章来自Google，作者还写过一篇ICA的Tutorial
- $y_k = \alpha_k^{\mathrm{T}} \boldsymbol{x}$考虑PCA是通过组合特征的方法来降维，这样用到**线性组合**。因为涉及到线性组合，所以在PCA过程中首先要给数据规范化就好理解了，也比较好理解数据的"结构"这种说法。
- 书中有提到在实际问题中，不同变量可能有不同的量纲，直接求主成分有时会产生不合理的结果。**消除这个影响**常对各个随机变量实施规范化，使其均值为0，方差为1。
- 关于主成分的性质，规范化的变量总体主成分主要是围绕特征值和特征向量展开的。
- 关于总体和样本的说明可以参考一下Strang的书[5]中第十二章部分说明。
- 关于k的选择，2000年有一个文章自动选择[3]。

内容

总体主成分分析

总体主成分性质

1. $\operatorname{cov}(\boldsymbol{y}) = \Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_m)$
2. $\sum\limits_{i=1}^{m} \lambda_i = \sum\limits_{i=1}^{m} \sigma_{ii}$
3. $\sum\limits_{i=1}^{m} \operatorname{var}(x_i) = \operatorname{tr}(\Sigma^{\mathrm{T}}) = tr(A\Lambda A^{\mathrm{T}}) = \operatorname{tr}(\Lambda) = \sum\limits_{i=1}^{m} \lambda_i = \sum\limits_{i=1}^{m} \operatorname{var}(y_i)$

两个拉格朗日函数的求导

规范化变量的总体主成分

这部分内容描述了规范化随机变量的总体主成分的性质，概括下就是：特征值，特征值的和，特征变量，特征变量按行求和，特征变量按列求和。

1. $\Lambda^* = \operatorname{diag}(\lambda_1^*, \lambda_2^*, \cdots, \lambda_m^*)$
2. $\sum\limits_{k=1}^{m} \lambda_k^* = m$
3. $\rho(y_k^*, x_i^*) = \sqrt{\lambda_k^*} e_{ik}^*, k, i = 1, 2, \cdots, m$
4. $\sum\limits_{i=1}^{m} \rho^2(y_k^*, x_i^*) = \sum\limits_{i=1}^{m} \lambda_k^* e_{ik}^{*2} = \lambda_k^*, k = 1, 2, \cdots, m$
5. $\sum\limits_{k=1}^{m} \rho^2(y_k^*, x_i^*) = \sum\limits_{k=1}^{m} \lambda_k^* e_{ik}^{*2} = 1, i = 1, 2, \cdots, m$

样本主成分分析

观测数据上进行主成分分析就是样本主成分分析。

给定样本矩阵X，可以**估计**样本均值以及样本协方差。

$\bar{x} = \frac{1}{n} \sum\limits_{j=1}^{n} x_j$

相关矩阵的特征值分解算法

关键词：**相关矩阵**， **特征值分解**

1. 观测数据规范化处理，得到规范化数据矩阵X
2. 计算相关矩阵R

$$R=[r_{ij}]_{m\times m}=\frac{1}{n-1}XX^{\mathrm{T}}$$
$$r_{ij}=\frac{1}{n-1}\sum_{l=1}^{n}x_{il}x_{lj},i,j=1,2,\cdots,m$$

3. 求R的特征值和特征向量

$$|R-\lambda I|=0$$
$$\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_m$$

求累计方差贡献率达到预定值的主成分个数k

$$\sum_{i=1}^{k}\eta_i=\frac{\sum\limits_{i=1}^{k}\lambda_i}{\sum\limits_{i=1}^{m}\lambda_i}$$

求前k个特征值对应的单位特征向量

$$a_i=(a_{1i},a_{2i},\cdots,a_{mi})^{\mathrm{T}}$$

4. 求k个样本主成分

$$y_i=a_i^{\mathrm{T}}\boldsymbol{x}$$

其实算法到这就完事了，剩下两部分是输出。**前面是fit部分，后面是transform部分。**具体可以看下P_{319}中的关于相关矩阵特征值分解算法部分内容，构造正交矩阵之后就得到了主成分。

5. 计算k个主成分y_i与原变量x_i的相关系数$\rho(x_i,y_i)$以及k个主成分对原变量x_i的贡献率ν_i

$$\nu_i=\rho^2(x_i,(y_1,y_2,\cdots,y_k))=\sum_{j=1}^{k}\rho^2(x_i,y_j)=\sum_{j=1}^{k}\lambda_j a_{ij}^2$$
$$i=1,2,\cdots,m$$

6. 计算n个样本的k个主成分值
   第j个样本，$\boldsymbol{x}_j=(x_{1j},x_{2j},\cdots,x_{mj})^{\mathrm{T}}$的第$i$个主成分值是

$$y_{ij}=(a_{1i},a_{2i},\cdots,a_{mi})(x_{1j},x_{2j},\cdots,x_{mj})^{\mathrm{T}}=\sum_{l=1}^{m}a_{li}x_{lj}$$
$$i=1,2,\cdots,m,j=1,2,\cdots,n$$

数据矩阵的奇异值分解算法

关键词：**数据矩阵**，**奇异值分解**
算法16.1 主成分分析法
输入：$m\times n$样本矩阵X，每一行元素均值为0。这里每一行是一个特征
输出：$k\times n$样本主成分矩阵Y
参数：主成分个数k

1. 构造新的$n\times m$矩阵

$$X'=\frac{1}{\sqrt{n-1}}X^{\mathrm{T}}$$

X'每一列均值为0，其实就是转置了。

2. 对矩阵X'进行截断奇异值分解

$$X' = U\Sigma V^{\mathrm{T}}$$

矩阵V的前k列构成k个样本主成分

3. 求$k \times n$样本主成分矩阵

$$Y = V^{\mathrm{T}} X$$

例16.1

这个例子，其实从表16.3中拿到的结论通过表16.2也能拿到。就是说通过单位特征向量和主成分的方差贡献率可以得到通过主成分的因子负荷量以及贡献率能得到的结论。
y_1是原始特征的线性组合，并且，各个原始特征的权重(系数)基本相同，说明大家同样重要。y_1和总成绩有关系。
y_2的贡献可能更多的体现在文理科的差异上，他们的作用相反。

类型	主成分	特征值	x_1	x_2	x_3	x_4	方差贡献率	备注
特征向量	y_1	2.17	0.460	0.476	0.523	0.537	0.543	
特征向量	y_2	0.87	0.574	0.486	-0.476	-0.456	0.218	累计0.761
因子负荷量	y_1	2.17	0.678	0.701	0.770	0.791	$\sqrt{\lambda_1}e_{i1}$	平方和=2.169
因子负荷量	y_2	0.87	0.536	0.697	0.790	0.806	$\sqrt{\lambda_2}e_{i2}$	平方和=0.870

这部分数值参考书上内容，如果用numpy做，会有一定出入，回头再复核下。

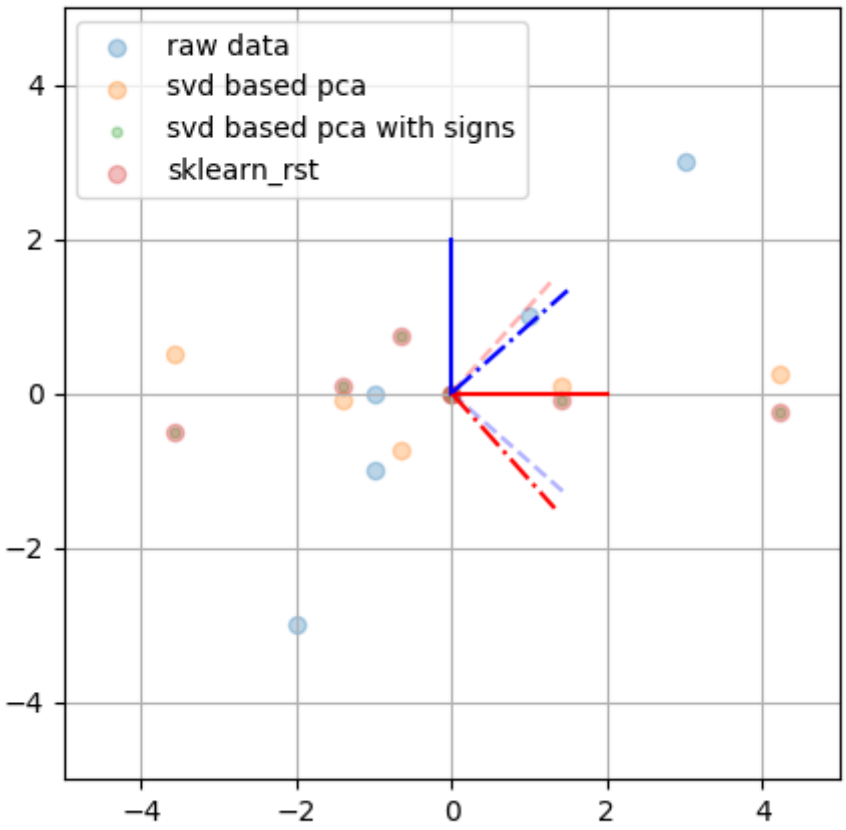
习题16.1

样本数据主成分分析

$$X = \begin{bmatrix} 2 & 3 & 3 & 4 & 5 & 7 \\ 2 & 4 & 5 & 5 & 6 & 8 \end{bmatrix}$$

这个题，原来就俩特征，然后主成分依然俩特征。俩特征就可以可视化了。

1. 首先要规范化，参考16.33，注意，规范化并不代表数据取值范围在$[0,1]$之间。



这里总共就两个特征，而且从数据的范围上看，差不多。

参考

CH17 潜在语义分析

前言

章节目录

1. 单词向量空间与话题向量空间
    1. 单词向量空间
    2. 话题向量空间
2. 潜在语义分析算法
    1. 矩阵奇异值分解算法
    2. 例子
3. 非负矩阵分解算法
    1. 非负矩阵分解
    2. 潜在语义分析模型
    3. 非负矩阵分解的形式化
    4. 算法

导读

- 潜在语义分析主要用于文本的话题分析，通过矩阵分解发现文本与单词之间的**基于话题**的语义关系。
- 词向量通常是稀疏的，词向量不考虑同义性，也不考虑多义性。
- 一个文本(Doc)一般有多个话题(Topic)。涉及到语义分析，要清楚什么是文本，什么是话题，什么是伪文本。
- NMF那个文章[参考文献3]发的是Nature，1999年的，不过他引不高，才9979。文章中对比了在矩阵分解框架下的VQ，PCA和NMF，说明了NMF和其他两种方法的区别。
- NMF的推导过程见参考文献4
- 潜在语义分析使用的是**非概率**的话题分析模型。
- 潜在语义分析是**构建话题向量空间的方法**(话题分析的方法)
- 单词向量转化成话题向量。文本在不同空间下的相似度用在不同空间下的向量内积表示。
- 话题向量空间T，单词-话题矩阵T，文本在话题空间的表示Y，话题-文本矩阵Y
- 本章第一个参考文献做了很多的文字说明，也有个实际的例子，可以参考下。
- 所谓**表示**，可以认为是在某个坐标系(空间)下的坐标。
- 非负矩阵分解旨在用较少的基向量、系数向量来表示较大的数据矩阵。
- 感觉这章的章节结构看起来不是很清晰，在内容部分重新梳理了下结构。
- 在sklearn中LSA就是截断奇异值分解，作为一种降维的手段进行处理。而NMF是单独的一个模型，都是矩阵分解的范畴。

内容

向量空间模型

单词向量空间

每个向量对应一个文本，单词向量空间行对应单词，话题向量空间行对应话题。
单词-文本矩阵

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

元素x_{ij}代表单词w_i在文本d_j中出现的频数或者权值。
X可以写作$X=\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$

单词多，文本少，这个矩阵是稀疏矩阵。

权值通常用TFIDF

$$TFIDF_{ij}=\frac{tf_{ij}}{tf_{\cdot j}}\log\frac{\mathrm{d}f}{\mathrm{d}f_i}$$
$$i=1,2,\cdots,m;$$
$$j=1,2,\cdots,n$$

一个单词在一个文本中的TFIDF是两种重要度的乘积，表示综合重要度。

话题向量空间

每个话题由一个定义在单词集合W上的m维向量表示，称为**话题向量**。
$t_l=\begin{bmatrix} t_{1l} & t_{2l} & \cdots & t_{ml} \end{bmatrix}^{\mathrm{T}}, l=1,2,\cdots,k$

k个话题向量张成一个话题向量空间，维数为k。

$$T=\begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1k} \\ t_{21} & t_{22} & \cdots & t_{2k} \\ \vdots & \vdots & & \vdots \\ t_{m1} & t_{12} & \cdots & t_{mk} \end{bmatrix}$$

矩阵T可以写成$T = \begin{bmatrix} t_1 & t_2 & \cdots & t_k \end{bmatrix}$

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & & \vdots \\ y_{k1} & y_{k2} & \cdots & y_{kn} \end{bmatrix}$$

矩阵Y可以写做$Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}$

$$x_j \approx y_{1j}t_1 + y_{2j}t_2 + \cdots + y_{kj}t_k, j = 1, 2, \cdots, n$$

这样，单词-文本矩阵X可以近似的表示为单词-话题矩阵T与话题-文本矩阵Y的乘积形式。这就是潜在语义分析。

$$X \approx TY$$

基于SVD的潜在语义分析模型

单词-文本矩阵

文本集合$D = \{d_1, d_2, \cdots, d_n\}$
单词集合$W = \{w_1, w_2, \cdots, w_m\}$

表示成单词-文本矩阵$X_{m \times n}$

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

截断奇异值分解

$$X \approx U_k \Sigma_k V_k^{\mathrm{T}} = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^{\mathrm{T}} \\ v_2^{\mathrm{T}} \\ \vdots \\ v_k^{\mathrm{T}} \end{bmatrix}$$

这中间$k \leq n \leq m$ 这里假设了文档数量要比单词数量少，其实这个假设也不一定成立。

1. U_k是$m \times k$矩阵，前k个相互正交的左奇异向量
2. Σ是k阶方阵，前k个最大奇异值
3. V_k是$n \times k$矩阵，前k个相互正交的右奇异向量

话题空间向量

每一列u_l表示一个话题，k个话题张成一个子空间，称为话题向量空间。
$U_k = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix}$

文本的话题空间向量表示

如果u_l表示话题向量空间，那么将文本表示成u_l的线性组合，就是文本在这个空间的表示。
但是，奇异值分解得到三个矩阵，最左边的是话题向量空间，那么右边的两个矩阵的乘积，则对应了话题-文本矩阵(文本的话题空间向量表示)。

这里有个点

$$V^{\mathrm{T}}=\begin{bmatrix} v_{11} & v_{21} & \cdots & v_{n1} \\ v_{12} & v_{22} & \cdots & v_{n2} \\ \vdots & \vdots & & \vdots \\ v_{1k} & v_{2k} & \cdots & v_{nk} \end{bmatrix}$$

问题：这个矩阵是$k \times n$的，右下角标感觉应该是v_{kn}这种形式？
这个矩阵是V^{T}，是k个特征值对应的特征向量做了归一化之后的结果，参考P_{258}中相应的描述，$A^{\mathrm{T}}A$的特征向量构成正交矩阵V的列。V是右奇异向量。
这就是为什么这个矩阵下标如此表示。

$$\begin{aligned} x_j &\approx U_k(\Sigma_k V_k^{\mathrm{T}})_j \\ &= \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix} \begin{bmatrix} \sigma_1 v_{j1} \\ \sigma_2 v_{j2} \\ \vdots \\ \sigma_k v_{jk} \end{bmatrix} \\ &= \sum_{l=1}^{k} \sigma_l v_{jl} u_l, j = 1, 2, \cdots, n \end{aligned}$$

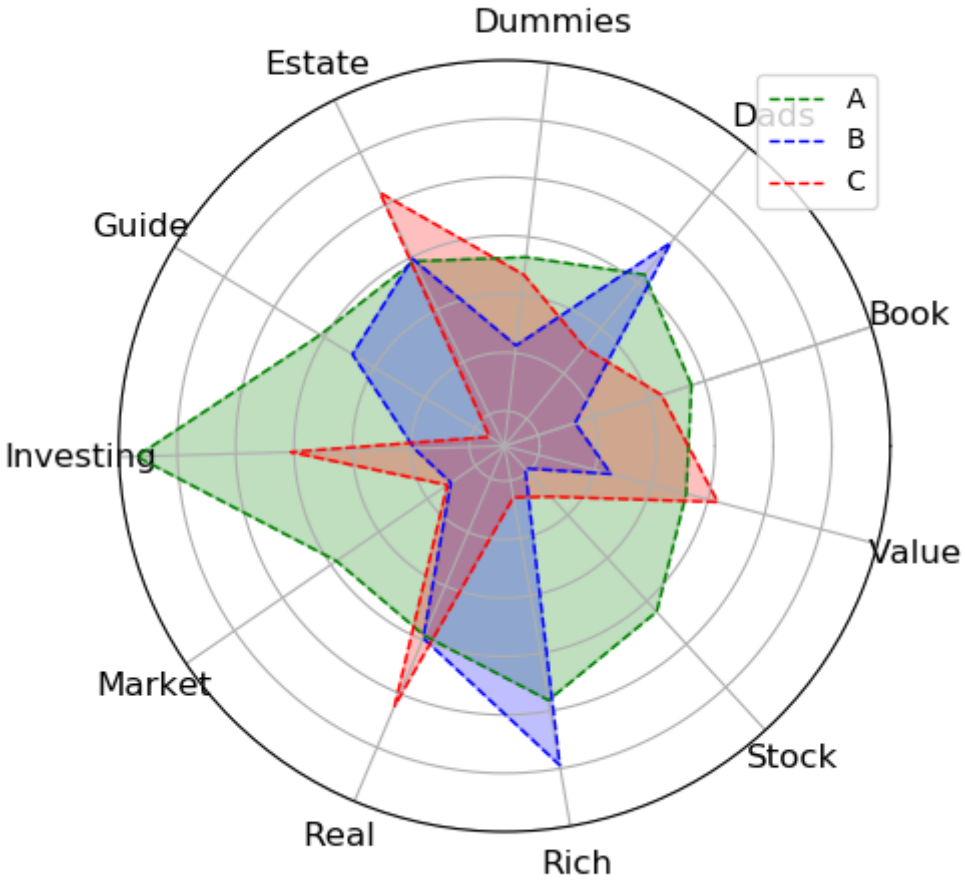
上式是文本d_j的近似表达式，由k个话题向量u_l的线性组合构成。
矩阵$(\Sigma_k V_k^{\mathrm{T}})$的每一个列向量是一个文本在话题向量空间的表示。

例子

书中这个例子原始数据是这样的：

1. The Neatest Little Guide to Stock Market Investing
2. Investing For Dummies, 4th Edition
3. The Little Book of Common Sense Investing: The Only Way to Guarantee Your Fair Share of Stock Market Returns
4. The Little Book of Value Investing
5. Value Investing: From Graham to Buffett and Beyond
6. Rich Dad's Guide to Investing: What the Rich Invest in, That the Poor and the Middle Class Do Not!
7. Investing in Real Estate, 5th Edition
8. Stock Investing For Dummies
9. Rich Dad's Advisors: The ABC's of Real Estate Investing: The Secrets of Finding Hidden Profits Most Investors Miss

去了停用词之后，做词频统计，得到了数据表。这个数据在概率潜在语义分析部分的习题中再次引用了。
对应的这部分数据，实际上还可以做一些事情。可以尝试可视化下。



上图中三个话题ABC，和不同单词的关系可以可以看出。也可以绘制单词-话题的雷达图。

这个例子里面书中给出的参考结果是按照V做了符号调整，保证V中每一行的最大值，符号为正。

基于NMF的潜在语义分析模型

NMF

X是非负矩阵则表示为$X \geq 0$
$X \approx WH, W \geq 0, H \geq 0$ 称为非负矩阵分解
非负矩阵分解旨在通过较少的基向量、系数向量来表达较大的数据矩阵。注意这里用到了基向量和数据矩阵，因为这部分内容介绍的是非负矩阵分解，和话题向量空间以及文本在话题向量空间的表示这些还没有联系在一起，是一个抽象的数学描述。

模型定义

$m \times n$的非负矩阵$X \geq 0$。
假设文本集合包含k个话题，对X进行非负矩阵分解。即求$m \times k$的非负矩阵和$k \times n$的非负矩阵满足$X \approx WH$

其中
$W = [w_1 \quad w_2 \quad \cdots \quad w_k]$表示话题向量空间
$w_1, w_2, \cdots, w_k$表示文本集合的k个话题
$H = [h_1 \quad h_2 \quad \cdots \quad h_k]$表示文本在话题向量空间的表示
$h_1, h_2, \cdots, h_k$表示文本集合的n个文本
以上是基于非负矩阵分解的潜在语义分析模型。

非负矩阵分解有很直观的解释，话题向量和文本向量都非负，对应着“伪概率分布”，向量的线性组合表示**局部构成总体**。这个其实和DL里面的意思是一样的。

算法

可以形式化为最优化问题求解。

损失函数

1. 平方损失
   两个非负矩阵$A=[a_{ij}]_{m\times n}$和$B=[b_{ij}]_{m\times n}$的平方损失定义为

$$\|A-B\|^2=\sum_{i,j}(a_{ij}-b_{ij})^2$$

下界是0

2. 散度

$$D(A\|B)=\sum_{i,j}\left(a_{ij}\log\frac{a_{ij}}{b_{ij}}-a_{ij}+b_{ij}\right)$$

下界是0
A和B不对称。
当$\sum\limits_{i,j}a_{ij}=\sum\limits_{i,j}b_{ij}=1$时散度损失函数退化为Kullback-Leiber散度或相对熵，这时A和B是概率分布。

问题定义

针对不同的损失函数有不同的问题定义

1. 平方损失

$$\min_{W,H}\|X-WH\|^2$$
$$s.t.W,H\geq 0$$

2. 散度损失

$$\min_{W,H}D(X\|WH)$$
$$s.t.W,H\geq 0$$

更新规则

这里提到目标函数只是对W和H之一的凸函数，而不是同时两个变量的凸函数，所以通过数值优化求解局部最优解。

1. 平方损失

$$H_{lj}\leftarrow H_{lj}\frac{(W^{\mathrm{T}}X)_{lj}}{(W^{\mathrm{T}}WH)_{lj}}$$
$$W_{il}\leftarrow W_{il}\frac{(XH^{\mathrm{T}})_{il}}{(WHH^{\mathrm{T}})_{il}}$$

2. 散度损失

$$H_{lj}\leftarrow H_{lj}\frac{\sum\limits_i[W_{il}X_{ij}/(WH)_{ij}]}{\sum\limits_i W_{il}}$$
$$W_{il}\leftarrow W_{il}\frac{\sum\limits_j[H_{lj}X_{ij}/(WH)_{ij}]}{\sum\limits_j H_{lj}}$$

NMF

1. 平方损失

$$J(W,H)=\frac{1}{2}\|X-WH\|^2=\frac{1}{2}\sum_{i,j}[X_{ij}-(WH)_{ij}]^2$$

采用梯度下降法求解
这里用到了矩阵求导

$$\frac{\partial J(W,H)}{\partial W_{il}} = -\sum_j [X_{ij} - (WH)_{ij}]H_{lj} = -[(XH^{\mathrm{T}})_{il} - (WHH^{\mathrm{T}})_{il}]$$
$$\frac{\partial J(W,H)}{\partial H_{lj}} = -[(W^{\mathrm{T}}X)_{lj} - (W^{\mathrm{T}}WH)_{lj}]$$

根据更新规则有

$$W_{il} = W_{il} + \lambda_{il}[(XH^{\mathrm{T}})_{il} - (WHH^{\mathrm{T}})_{il}]$$
$$H_{lj} = H_{lj} + \mu_{lj}[(W^{\mathrm{T}}X)_{lj} - (W^{\mathrm{T}}WH)_{lj}]$$
$$\lambda_{il} = \frac{W_{il}}{(WHH^{\mathrm{T}})_{il}}$$
$$\mu_{lj} = \frac{H_{lj}}{(W^{\mathrm{T}}WH)_{lj}}$$

算法

输入：单词-文本矩阵$X \geq 0$，文本集合的话题个数k，最大迭代次数t；
输出：话题矩阵W，文本表示矩阵H

1. 初始化
   $W \geq 0$，**并对W的每一列数据归一化**
   $H \geq 0$
2. 迭代
   对迭代次数从1到t执行下列步骤：
   a. 更新W的元素，每次迭代对W的列向量归一化，**使基向量为单位向量。**
   b. 更新H的元素

习题

习题18.3

参考

1. Maximum-likelihood from incomplete data via the EM algorithm↩↩↩
2. [Convex Combination](https://en.wikipedia.org/wiki/Convex_combination↩↩
3. 多元正态分布↩↩↩↩
4. Machine Learning，P_{618}↩↩↩
5. Gap Statistics↩↩↩↩↩