机器学习笔记

Yuxuan Yuan & Lulu Cao

2022.2.21

目录

1 Lecture 1

2022.2.21 第一节课是在 C103 上的，啥也没带 oh lambda 含义

2 Lecture 2

2022.2.28 yyx 忘了记录板书 oh

2.1 课堂回顾

机器学习 =lambda ：机器 = 函数；学习 = 拟合

Machine Learning = LAMBDA. Loss,Algorithm, Model, BigData, Application。

BigData $D = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$，其中 $x_n \in R^N$，$y \in R$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_N^{\mathrm{T}} \end{bmatrix}$$

Model

$$y = \boldsymbol{w^T x}$$

Loss

$$\mathcal{L}_2(\boldsymbol{w}) = \frac{1}{N}\sum_{n=1}^{N}\left(y_n - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n\right)^2$$

将上述式子矩阵化，

$$\begin{aligned}
\mathcal{L}_2(\boldsymbol{w}) &= \frac{1}{N}(\boldsymbol{Xw}-\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{Xw}-\boldsymbol{y}) \\
&= \frac{1}{N}\left((\boldsymbol{Xw})^{\mathrm{T}} - \boldsymbol{y}^{\mathrm{T}}\right)(\boldsymbol{Xw}-\boldsymbol{y}) \\
&= \frac{1}{N}(\boldsymbol{Xw})^{\mathrm{T}}\boldsymbol{Xw} - \frac{1}{N}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{Xw} - \frac{1}{N}(\boldsymbol{Xw})^{\mathrm{T}}\boldsymbol{y} + \frac{1}{N}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} \\
&= \frac{1}{N}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Xw} - \frac{2}{N}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + \frac{1}{N}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}
\end{aligned}$$

Algorithm 对损失函数求导，并令其等于 0

$$\begin{aligned}
\frac{\partial \mathcal{L}_2}{\partial w} &= \frac{2}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Xw} - \frac{2}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} = 0 \\
\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Xw} &= \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}
\end{aligned}$$

矩阵求导相关

$f(w)$	$\frac{\partial f}{\partial w}$
$w^{\mathrm{T}}\boldsymbol{x}$	$\boldsymbol{x}$
$\boldsymbol{x}^{\mathrm{T}}w$	$\boldsymbol{x}$
$w^{\mathrm{T}}w$	$2w$
$w^{\mathrm{T}}\boldsymbol{C}w$	$2\boldsymbol{C}w$

从而，

$$\boldsymbol{w} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

2.2 投影矩阵 & Normal Equation

2.3 增加新的数据 (样本数 or 特征维度)

作为作业，当 $\boldsymbol{X}$ 增加一行 (样本数增加) 或者增加一列 (特征维度增加) 时，$\boldsymbol{W}$ 如何变化，写出更新后的 $\boldsymbol{W}$ 和更新前的 $\boldsymbol{W}$ 之间的增量表达式。

相关公式

- **Sherman-Morrison 公式**

$$\left(A+uv^T\right)^{-1}=A^{-1}-\frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}$$

- **分块矩阵求逆**　设 A 是 $m\times m$ 可逆矩阵, B 是 $m\times n$ 矩阵, C 是 $n\times m$ 矩阵, D 是 $n\times n$ 矩阵, $D-CA^{-1}B$ 是 $n\times n$ 可逆矩阵, 则有

$$\left[\begin{array}{cc} A & B \\ C & D \end{array}\right]^{-1}=\left[\begin{array}{cc} A^{-1}+A^{-1}B\left(D-CA^{-1}B\right)^{-1}CA^{-1} & -A^{-1}B\left(D-CA^{-1}B\right)^{-1} \\ -\left(D-CA^{-1}B\right)^{-1}CA^{-1} & \left(D-CA^{-1}B\right)^{-1} \end{array}\right]$$

2.4 人工智能的流派

- 类比主义：核方法、SVM
- 连接主义：神经网络
- 贝叶斯主义
- 符号主义：决策树、专家系统
- 演化主义（优化算法）：遗传算法
- 行为主义：强化学习

2.5 人脸识别

设 D 为人脸的数据, $D=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^N$，其中 $x_n\in R^N$，$y\in[1,100]$, $y_i=w^Tx_i+\epsilon,\epsilon\sim W(0,\sigma^2)$, $\sigma^2=1$, 则 $y_i\sim W(w^Tx_i,1)$

方差一样，所以一样胖；均值不一样，所以 location 不同

找到一个 w，使得 y_i 出现的概率最大, 即 $P(y_i|w^Tx,1)=\frac{1}{\sqrt{2\pi}}exp{-\frac{1}{2}(\frac{y_i-w^Tx}{1})^2}$

$$L=p\left(\boldsymbol{y}\mid\boldsymbol{X},w,\sigma^2\right)=\prod_{n=1}^N p\left(y_n\mid\boldsymbol{x}_n,w,\sigma^2\right)=\prod_{n=1}^N\mathcal{N}\left(w^{\mathrm{T}}\boldsymbol{x}_n,\sigma^2\right)$$

取对数, 有

$$\begin{aligned}\log L&=\sum_{n=1}^N\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}\left(y_n-f\left(\boldsymbol{x}_n;\boldsymbol{w}\right)\right)^2\right\}\right)\\&=\sum_{n=1}^N\left(-\frac{1}{2}\log(2\pi)-\log\sigma-\frac{1}{2\sigma^2}\left(y_n-f\left(\boldsymbol{x}_n;w\right)\right)^2\right)\\&=-\frac{N}{2}\log 2\pi-N\log\sigma-\frac{1}{2\sigma^2}\sum_{n=1}^N\left(y_n-f\left(\boldsymbol{x}_n;w\right)\right)^2\end{aligned}$$

$w^{\mathrm{T}}\boldsymbol{x}_n$ 替换模型中的决定性部分, 对数似然表达式就呈现如下的形式: $\log L=-\frac{N}{2}\log 2\pi-N\log\sigma-\frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(y_n-w^{\mathrm{T}}\boldsymbol{x}_n\right)^2$ 得到的最小二乘解, 通过求导数、使其等于零以及求解拐点的方法, 类似于 1.1.4 节所述的方式。对于 w (注意, $w^{\mathrm{T}}\boldsymbol{x}_n=\boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{w}$),

最小二乘法和极大似然估计是等价的

3 Lecture 3

2022.3.7

Outline

1. Least Square: MLE
   1.1 SGD
   1.2 Probalistic Graph Representation
   1.3 $\mathbb{E}[\hat{w}]/cov[\hat{w}]$
   1.4 bias / variance
2. Revisted LS: Curve Fitting
   2.1 Model Selection
   2.2 Overfitting
3. How to solve overfitting
   3.1 Regularization (MAP)
   3.2 Bayesian Learning

4 Lecture 4

2022.3.14

使用 MLE/MAP/Bayes Learning 三种方法来求解参数，通过 4 个例子来加深。Example 4 没讲完。其中 Example 3 的三种解法需要自己课后补充。

tips: to be added 三个图对应生成式模型、判别式模型、分布计算; 一些前置 discrete continuous 的概率表变量分布；有积分的地方和是求谁的期望的地方，

Outline

- MLE / MAP / Bayesian
- Example 1. Bernoulli
- Example 2. Gaussian
- Example 3. Linear Regression
- Example 4. Logistic Regression

问题:

- 共轭分布是什么意思
- β 分布

4.1 Example 1. Bernoulli

Given dataset $D = \{< x_i >\}_{i=1}^n, x_i \in \{0, 1\}$ x_i 服从 Bernoulli 分布，$P(x_i = 1) = \theta$; 再假设 n_1 表示 D 中 $x_i = 1$ 的个数；

Method

1. MLE

$$\begin{aligned} P(D|\theta) &= \prod_{i=1}^n P(x_i; \theta) \\ &= \prod_{i=1}^n \theta^{x_i}(1-\theta)^{1-x_i} \\ &= \theta^{\sum_i x_i}(1-\theta)^{n-\sum_i x_i} \\ &= \theta^{n_1}(1-\theta)^{n-n_1} \end{aligned}$$

取对数以便计算，从而

$$\arg\max_\theta \ln P(D|\theta) = \arg\max_\theta n_1 \ln\theta + (n-n_1)\ln(1-\theta)$$

对 θ 求导

$$\frac{n_1}{\theta} - \frac{n-n_1}{1-\theta} = 0$$

得到 $\theta = \frac{n_1}{n}$

2. MAP

Beta 分布 $Beta(\theta \mid a, b)$ $P(\theta|D) \propto P()$

贝塔分布（Beta Distribution) 是一个作为伯努利分布和二项式分布的共轭先验分布的密度函数。在概率论中，贝塔分布，也称 B 分布，是指一组定义在 (0,1) 区间的连续概率分布。

3. Bayesian

4.2 Example 2. Gaussian

Method

1. MLE
2. MAP
3. Bayesian

4.3 Example 3. Linear Regression

Given dataset $D = \{< x_i, y_i >\}_{i=1}^n, y_i \in \mathbb{R}$, 假设 $X\ \mathcal{N}(\theta, \sigma^2)$，则有 $P(x;\theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{(x-\theta)^2}{2\sigma^2}\}$

Method

1. MLE
2. MAP
3. Bayesian

4.4 Example 4. Logistic Regression

Method

1. MLE

2. MAP 3. Bayesian

计算步骤 (流程) 总结

1. $P(D|\theta)$
2. $P(\theta)$
3. $P(\theta|D) \propto P(D|\theta)P(\theta)$
4. $\theta_{bayes} = \mathbb{E}[\theta|D]$
5. inference: $P(y_{new}|D, x_{new}) = \int p(y_{new}|x_{new}, \theta)p(\theta|D)$

其中 $D = (X, Y)$

5 Lecture 5

2022.3.21

Outline

1. Review: MLE / MAP / Bayesian Estimation
2. Logistic Regression
3. Perceptron
4. Generative Classification Model

    x is continuous (GDA)

    x is discrete (NB)

....

Data: $D=<x_i,y_i>_{i=1}^n, x_i\in\mathbb{R}^d, y_i\in 0,1, D=(X,Y)$

Model: $P(x,y;\Theta)$(生成式) ; $P(y|x;\Theta)$(判别式) ; $P(x_i;\Theta)$ 无监督

Inference: Given x, D; Output y, $P(y|x,D)=?$

Learning: Given D; Output Θ, $P(D|\Theta)$, $\Theta_{Bayes}=\mathbb{E}[\Theta|D]$

....

5.1 Review: MLE / MAP / Bayesian

三种方法，即 Learning 的三种方法

Learning

1. MLE

$$\begin{aligned}\hat{\Theta}_{MLE}&=\arg\max_{\Theta}P(D|\Theta)\\&=\arg\max_{\Theta}\sum_{i=1}^{n}\ln P(blank;\Theta)\end{aligned}$$

其中，$blank$ 可以填入 1)x_i,y_i; 2)$y_i|x_i$; 3)x_i; 即三种模型都可用 MLE 求解

2. MAP

$$\begin{aligned}\hat{\Theta}_{MAP}&=\arg\max_{\Theta}P(\Theta|D)\propto P(D|\Theta)P(\Theta)\\&=\arg\max_{\Theta}\ln P(blank;\Theta)+\ln P(\Theta)\end{aligned}$$

其中，$P(\Theta)$ 是先验，$P(D|\Theta)$ 是似然，等式的最后 $\ln P(blank$ 为数据项（Data Term），$\ln P(\Theta)$ 为正则项 (Regularization Term, 或平滑项 (Smooth Term))。

* 当 $P(\Theta)$ 是均匀分布时，$MLE=MAP$ 。

3. Bayesian Estimation

(前提是 $P(\Theta|D)$ 即后验分布已知)

$$\hat{\Theta}_{Bayes}=\mathbb{E}[\Theta|D]=E_{\theta\sim P(\cdot|D)}[\Theta]=\int\Theta p(\Theta|D)d\Theta$$

Inference

利用上述三种方法做参数估计后的推理方法

1. MLE

Given $\hat{\Theta}_{MLE}, x, D$, ouput $P(y|x;\hat{\Theta}_{MLE})$.

举例，在 Logistic Regression 中，$P(y=1|x;\hat{\Theta}_{MLE}) = \sigma(\hat{\Theta}_{MLE}^T x)$

2. MAP

Given $\hat{\Theta}_{MAP}, x, D$, output $P(y|x;\hat{\Theta}_{MAP})$.

举例，在 Logistic Regression 中，$P(y=1|x;\hat{\Theta}_{MAP}) = \sigma(\hat{\Theta}_{MAP}^T x)$

3.Bayes Estimation

Given $x, D, P(\Theta|D)$, ouput $P(y|x;\hat{\Theta}_{MAP})$.

$$\begin{aligned} P(y|x;D) &= \int p(y,\Theta|x;D)d\Theta \\ &= \int p(\Theta|x;D)p(y|\Theta,x;D)d\Theta \end{aligned}$$

等式最后的 $p(\Theta|x;D)$ 为后验分布，$p(y|\Theta,x;D)$ 为模型。能这么做的前提是假设 x_{new} 与 Θ 无关。

如果求 $y=1$, 有

$$\begin{aligned} p(y|\Theta,x;D) &= \int p(\Theta|D)\sigma(\Theta^T x)dx (= \mathbb{E}_{\Theta\sim P(\cdot|D)}[\sigma(\Theta^T x)] \\ &\approx \frac{1}{\mathcal{L}}\sum_{i=1}^{\mathcal{L}} \sigma((\Theta^{(i)})^T x) \end{aligned}$$

其中有假设 x_{new} 与 Θ 相互独立；$\Theta^{(i)} \sim P(\Theta|D))), i = 1, ..., \mathcal{L}$, 上面公式使用了抽样技术 (Sampling) 来求期望。

5.2 Logistic Regression(逻辑斯蒂回归)

补点图概率图模型（可观测量用灰色阴影）

Learning

1.MLE

$$P(y_i|x_i;\Theta) = \sigma(\Theta^T x_i)^{y_i}(1-\sigma(\Theta^T x_i))^{1-y_i}$$

$$\begin{aligned} lnP(D|\Theta) &= \sum_{i=1}^{n}\{y_i \ln \sigma_i + (1-y_i)\ln(1-\sigma_i)\} \\ &= \mathcal{L}_D(\Theta) \end{aligned}$$

其中 $\sigma_i = \sigma(a_i) = \sigma(\Theta^T x_i) = \frac{1}{1+e^{\Theta^T x_i}}$，从而

$$\Theta^* = \arg\max_{\Theta} \mathcal{L}_D(\Theta)$$

到这里，求解 Θ^* 方法有求偏导数并等于 0 来计算解析解，但无法做到，原因如下, 利用链式法则算一下偏导数

$$\begin{aligned}\nabla_\Theta \mathcal{L}_D(\Theta) &= \frac{\partial \mathcal{L}_D(\Theta)}{\partial \Theta} \\ &= \frac{\partial \mathcal{L}_D(\Theta)}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial a_i}\frac{\partial a_i}{\partial \Theta} \\ &= \sum_{i=1}^{n}(\frac{y_i}{\sigma_i} - \frac{1-y_i}{1-\sigma_i})\sigma_i(1-\sigma_i)x_i \\ &= \sum_{i=1}^{n}(y_i - \sigma_i)x_i\end{aligned}$$

试试，几乎无法求得解析解。

第二种方法，尝试二阶导数 $\nabla_\Theta(\nabla_\Theta \mathcal{L}_D(\Theta))$，令 $\nabla_\Theta \mathcal{L}_D(\Theta) = g$,

Algorithm 1 A1: Gradient Ascent for Logistic Regression

Input: X, Y;

Output: Θ;

init: $\Theta \sim \mathcal{N}(0, \alpha^{-1}\mathrm{I}), \epsilon$

Loop:

  $g = \mathbf{X}^T(\mathbf{Y} - \sigma)$

  $\Theta^{t+1} := \Theta^t + \eta g$

until $|| \ \Theta^{t+1} - \Theta^t \ || \leq \epsilon$

return Θ;

Hessian Matrix of the Loss Function $\mathcal{L}_D(\Theta)$

Newton $\Theta^{t+1} := \Theta^t + \mathrm{H}^{-1}g$ ......todo

2.MAP

$$P(\Theta) \sim \mathcal{N}(0, \alpha^{-1}\mathrm{I})$$

$$P(D|\Theta) = \prod_{i=1}^{n} \sigma_i^{y_i}(1-\sigma_i)^{1-y_i}$$

$$P(\Theta|D) \propto \mathcal{N}(0, \alpha^{-1}\mathrm{I}) \prod_{i=1}^{n} \sigma_i^{y_i}(1-\sigma_i)^{1-y_i}$$

根据 MAP，有

$$\begin{aligned}\Theta^* &= \arg\max_\Theta \ln P(\Theta|D) \\ &= \arg\max_\Theta \ln P(D|\Theta) + \ln P(\Theta) \\ &= \arg\max_\Theta \sum_{i=1}^{n} y_i \ln(\sigma_i) + (1-y_i)\ln(1-\sigma_i) - \frac{1}{2}(\Theta^{-1}\Sigma^{-1}\Theta)\end{aligned}$$

则

$$\nabla_\Theta \mathcal{L}_D(\Theta) = \frac{\partial \mathcal{L}_D(\Theta)}{\partial \Theta} = \sum_{i=1}^{n}(y_i - \sigma_i)x_i - \Theta\Sigma^{-1}$$

这里定 $\alpha^{-1}\mathrm{I} = \Sigma$, 不影响结果。

类似算法A1, 写个A2

Algorithm 2 A2: (MAP) Gradient Ascent for Logistic Regression

Input: X, Y;

Output: Θ;

init: $\Theta \sim \mathcal{N}(0, \alpha^{-1}\mathrm{I}), \epsilon$

Loop:

  $g = \mathbf{X}^T(\mathbf{Y} - \sigma) - \Theta^t\Sigma^{-1}$

  $\Theta^{t+1} := \Theta^t + \eta g$

until $|| \Theta^{t+1} - \Theta^t || \leq \epsilon$

return Θ;

3.Bayesian Estimation

贝叶斯估计：$\mathbb{E}[\Theta|D]$

$P(y = 1|x, D) =$ 类似算法A1, 写个A3用于贝叶斯推理

Inference 后验预测分布为

$$p(y = 1|x_{new}, D) = \int p(y = 1|x_{new}, \Theta)p(\Theta|x, D)d\Theta$$

但是积分难以处理，采用近似处理，同时还有假设 x_{new} 与 Θ 无关，有

$$p(y = 1|x_{new}, D) = \int p(y = 1|x_{new}, \Theta, D)p(\Theta|D)d\Theta = \int \sigma(\Theta^T x_{new})p(\Theta|D)d\Theta \approx \frac{1}{\mathcal{L}}\sum_{i=1}^{\mathcal{L}} \sigma((\Theta^{(i)})^T x_{new})$$

其中，$\Theta^{(i)} \propto p(\Theta|D)(i = 1, ..., \mathcal{L})$

Algorithm 3 A3: Inference after Bayesian Estimation

Input: x_{new}, D;

Output: y;

init: $\Theta^{(i)} \propto p(\Theta|D)(i = 1, ..., \mathcal{L})$

$P(y = 1|x_{new}, D) = \frac{1}{\mathcal{L}}\sum_{i=1}^{\mathcal{L}} \sigma((\Theta^{(i)})^T x_{new})$

$P(y = 0|x_{new}, D) = \frac{1}{\mathcal{L}}\sum_{i=1}^{\mathcal{L}} (1 - \sigma((\Theta^{(i)})^T x_{new}))$

return 1 if $P(y = 1|x_{new}, D) > P(y = 0|x_{new}, D)$ else 0;

5.3 Perceptron(感知机)

写个历史表

- 1943 M-P —-神经元模型
- 1957 Rosenblatt —-Perceptron;
- 1982 BP 算法 (Computational Graph 计算图)
- 2006 Hinton —-预训练 pretain、DNN
- 2012 AlexNet.... 图像不太懂...

Model
计算方法：$y = sgn(\mathbf{W}^T x + b)$
其中;

$$\begin{cases} y = +1, \mathbf{W}^T x + b > 0, \\ y = -1, \mathbf{W}^T x + b < 0 \end{cases}$$

Loss 计算

$$\mathcal{L}_D(W) = \sum_{x_i, y_i \in M} \frac{|y_i(W^T x_i + b)|}{||W||}$$

其中 $M = \{(x_i, y_i) | y_i(W^T x + b) < 0\}$，可令 $||W|| = 1$，上式可写成,

$$\mathcal{L}_D(W) = \sum_{i=1}^{n} \max\{0, -y_i(W^T x_i + b)\}$$

* 当 max 中的第二项变为 $1 - y_i(W^T x_i + b)$ 时，损失函数变成 SVM 的损失函数。
这里求一下偏导数, 对于单个样本 (x_i, y_i)

$$\frac{\partial(-y_i(W^T x_i + b))}{\partial W} = -y_i x_i$$

也可以写成一个算法形式

Algorithm 4 A4: Perceptron GD

Input: X, Y;
Output: $\Theta = \{W, b\}$;
init: W, b
Loop:
  if $y_i(W^T x_i + b) < 0$ **then**
    $W^{t+1} \leftarrow W^t - \eta \nabla_W \mathcal{L}_D(W)$
    等价于 $W^{t+1} \leftarrow W^t + \eta y_i x_i$
  end if
until 训练集中没有误分类点
return Θ;

5.4 Generative Classification Model

y 是离散的

x 是连续的

Given $\{(x_i,y_i)\}_{i=1}^n = D = (X,Y), x \in \mathbb{R}^d$。假设 $y_i = 0, 1$，即 y_i 服从 Bernoulli 分布，x_i 服从高斯分布 $x_i|y_i \sim \mathcal{N}(\mu_k, \Sigma_k)$，model 描述如下

$$\begin{aligned}P(x_i, y_i; \Theta) &= P(y_i;\Theta)P(x_i|y_i;\Theta) \\ &= \text{Bernoulli}(y_i|p)\prod_{y=0}^{1}\mathcal{N}(x_i|\mu_k,\Sigma_k)^{\mathbf{1}\{k=y\}}\end{aligned}$$

关于 $P(x_i|y_i;\Theta)$ 部分细述如下,

$$\begin{aligned}P(x|y=1;\Theta) &= \mathcal{N}(x|\mu_1,\Sigma_1) \\ P(x|y=0;\Theta) &= \mathcal{N}(x|\mu_0,\Sigma_0)\end{aligned}$$

从而有

$$P(x|y;\Theta) = \mathcal{N}(x|\mu_1,\Sigma_1)^{\mathbf{1}\{y=1\}}\mathcal{N}(x|\mu_0,\Sigma_0)^{\mathbf{1}\{y=0\}}$$

记 $\Theta = (p, \mu_0, \Sigma_0, \mu_1, \Sigma_1)$，则有

$$\begin{aligned}P(D;\Theta) &= \prod_{i=1}^{n} P(x_i, y_i;\Theta) \\ &= \prod_{i=1}^{n}(p^{y_i}(1-p)^{1-y_i})\mathcal{N}(x|\mu_1,\Sigma_1)^{\mathbf{1}\{y=1\}}\mathcal{N}(x|\mu_0,\Sigma_0)^{\mathbf{1}\{y=0\}}\end{aligned}$$

取对数，有

$$\ln P(D;\Theta) = \sum_{i=1}^{n} y_i \ln p + (1-y_i)\ln(1-p) + \sum_{i=1}^{n}\mathbf{1}\{y=1\}\ln\mathcal{N}(x_i|\mu_1,\Sigma_1) + \sum_{i=1}^{n}\mathbf{1}\{y=0\}\ln\mathcal{N}(x_i|\mu_0,\Sigma_0)$$

Learning time !!!!! **MLE**

根据 $\hat{\Theta}_{MLE} = \arg\max_{\Theta}\ln P(D;\Theta)$，假设 Σ_0, Σ_1 已知，设 $M_1 = \{(x_i,y_i)|y_i = 1\}, M_0 = \{(x_i,y_i)|y_i = 0\}$，求 p, μ_0, μ_1。

$$\begin{aligned}\mathcal{L}_D(\Theta) &= \ln P(D;\Theta) \\ \ln\mathcal{N}(x|\mu,\Sigma) &= -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) \\ \frac{\partial\mathcal{L}_D(\Theta)}{\partial p} &= \sum_{i=1} w^n(\frac{y_i}{p} - \frac{1-y_i}{1-p}) = 0 \\ \frac{\partial\mathcal{L}_D(\Theta)}{\partial\mu_0} &= \sum_{(x_i,y_i)\in M_0}\Sigma^{-1}(x_i-\mu_0) = 0 \\ \frac{\partial\mathcal{L}_D(\Theta)}{\partial\mu_1} &= \sum_{(x_i,y_i)\in M_1}\Sigma^{-1}(x_i-\mu_1) = 0\end{aligned}$$

解得

$$p = \frac{\sum_{i=1}^{n} y_i}{n}$$
$$\mu_0 = \frac{\sum_{(x_i,y_i)\in M_0} x_i}{|M_0|}$$
$$\mu_1 = \frac{\sum_{(x_i,y_i)\in M_1} x_i}{|M_1|}$$

Infering time !!!

Given $x, y = ?$， $p, \mu_0, \mu_1, \Sigma_0, \Sigma_1$ are known.

$$\begin{aligned} P(y=1|x) &\propto P(y=1)P(x|y=1) \\ &= p\mathcal{N}(x|\mu_1, \Sigma_1) \\ P(y=0|x) &\propto P(y=0)P(x|y=0) \\ &= (1-p)\mathcal{N}(x|\mu_0, \Sigma_0) \end{aligned}$$

若 $P(y=1|x) > P(y=0|x)$ ，则 $y=1$。

5.5 作业

$$\begin{aligned} g(x) &= \ln \frac{P(y=1|x)}{P(y=0|x)} \\ &= \ln \frac{p\mathcal{N}(x|\mu_1, \Sigma_1)}{(1-p)\mathcal{N}(x|\mu_0, \Sigma_0)} \end{aligned}$$

设 $\Sigma_1 = \Sigma_0$, 则 $g(x)$ 是线性平面, 可写作 $g(x) = w^T x + b$, 即 Gaussian Discriminative Analysis(GDA), 求 w, b

$$\begin{aligned} g(x) &= \ln \frac{p}{1-p} + (\frac{1}{2}((x-\mu_0)^T \Sigma^{-1}(x-\mu_0) - (x-\mu_1)^T \Sigma^{-1}(x-\mu_1))) \\ &= \ln \frac{p}{1-p} + \frac{1}{2}((x^T - {\mu_0}^T)\Sigma^{-1}(x-\mu_0) - (x^T - {\mu_1}^T)\Sigma^{-1}(x-\mu_1)))) \\ &= \ln \frac{p}{1-p} + \frac{1}{2}(x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_0 - {\mu_0}^T\Sigma^{-1}x + {\mu_0}^T\Sigma^{-1}\mu_0 + x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_1 + {\mu_1}^T\Sigma^{-1}x - {\mu_1}^T\Sigma^{-1}\mu_1) \\ &= \ln \frac{p}{1-p} + \frac{1}{2}(2({\mu_1}^T\Sigma^{-1}x - {\mu_0}^T\Sigma^{-1}x) + {\mu_0}^T\Sigma^{-1}\mu_0 - {\mu_1}^T\Sigma^{-1}\mu_1) \\ &= (\mu_1 - \mu_0)^T\Sigma^{-1}x + \frac{1}{2}{\mu_0}^T\Sigma^{-1}\mu_0 - \frac{1}{2}{\mu_1}^T\Sigma^{-1}\mu_1 + \ln p - \ln(1-p) \end{aligned}$$

即 (结合：w 记得算一下转置，Σ 是对称矩阵等内容)

$$\begin{aligned} w &= \Sigma^{-1}(\mu_1 - \mu_0) \\ b &= \frac{1}{2}{\mu_0}^T\Sigma^{-1}\mu_0 - \frac{1}{2}{\mu_1}^T\Sigma^{-1}\mu_1 + \ln p - \ln(1-p) \end{aligned}$$

x 是离散的

6 Lecture 6

2022.3.28

Outline

1. Generative Classification Model : $p(x, y)$, y is discrete

   x is continuous (高斯判别分析 GDA)

   x is discrete (朴素贝叶斯 NB)

2. 贝叶斯学习

6.1 朴素贝叶斯 NB

Problem: $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$，其中 $x_i \in R^d$，$y_i \in \{0, 1\}$, $< X, Y >\sim P$

6.1.1 原理与模型

假设：各个维度独立；每个维度的取值为 0 或 1

$$P(X_1, ..., X_d|Y) = \prod_{j=1}^{d} P(X_j|Y)$$

条件独立：$P(X|Y, Z) = P(X|Z) = X \perp Y|Z$

$P(X|Y) = P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y) = P(X_1|Y)P(X_2|Y)$

模型：

$$\begin{aligned} y^* &= \arg\max_{y_k} \frac{P(y_k)P(X|y_k)}{P(X)} \\ &= \arg\max_{y_k} P(Y = y_k) \prod_{j=1}^{d} P(X_j = x_j|Y = y_k) \end{aligned}$$

6.1.2 算法

1. MLE

   其实就是计算频率

2. 零频率问题

   解决方案：拉普拉斯平滑

   $$P(w \mid c) = \frac{\mathrm{num}(w, c) + \varepsilon}{\mathrm{num}(c) + k\varepsilon}$$

   PS: 属性缺失不用管

6.1.3 小结：产生式 p(y|x) vs. 判别式 p(x,y)

1. 产生式模型有其对应的判别式模型，这样的组合叫做“产生式-判别式”对；(e.g. 根据 p(x,y) 得到 GDA 的分界面);
2. 判别式模型不一定有其对应的产生式模型;
3. 训练样本多的情况下，采用判别式;
4. 训练样本少的情况下，选择产生式。

6.2 贝叶斯学习

1. MLE/MAP/贝叶斯估计
2. 贝叶斯估计：将先验信息集成到模型推理中 $p(x, X, \theta)$

作业

朴素贝叶斯的贝叶斯估计

输入: 训练集 $T=\{(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)\}$, 其中 $x_i=\left(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}\right)^T$,第 i 个样本的第 j 个特征 $x_i^{(j)}$ 可能的取值集合为 $\{a_{j1}, a_{j2}, \ldots, a_{jS_j}\}$ ，即某一特征的取值为有限集合。$y_i \in \{c_1, c_2, \ldots, c_K\}$ ，即分类结果为有限集合 (共 K 个)。

先验概率的贝叶斯估计为:

$$P(Y=c_k)=\frac{\sum_{i=1}^N I(y_i=c_k)+\lambda}{N+K\lambda}, k=1,2,\ldots,K$$

条件概率的贝叶斯估计为:

$$P\left(X^{(j)}=a_{jl} \mid Y=c_k\right)=\frac{\sum_{i=1}^N I\left(x_i^{(j)}=a_{jl}, y_i=c_k\right)+\lambda}{\sum_{i=1}^N I(y_i=c_k)+S_j\lambda}$$

朴素贝叶斯的贝叶斯估计为:

$$y=\arg\max_{c_k} P(Y=c_k)\prod_{j=1}^n P(X^{(j)}=x_j \mid Y=c_k)$$

7 Lecture 7

2022.4.4

Outline

1. 狄利克雷分布
2. MLE/MAP/Bayesian Learning：多分类
3. Bernoulli Mixture Model
4. EM 算法

   EM-principle

   EM-convergence

   EM for GMM

   EM for Topic Model

7.1 狄利克雷分布

7.1.1 二项分布

二项分布用以描述 n 次独立的伯努利实验中有 m 次成功的概率:

$$P(X=m)=\binom{n}{m}p^m(1-p)^{n-m}$$

当试验的次数 n 为 1 时，二项分布变成伯努利分布。

7.1.2 多项分布

多项分布是一种多元离散随机变量的概率分布，是二项分布的扩展。假设重复进行 n 次独立随机试验，每次试验可能出现的结果有 k 种，第 i 种结果出现的概率为 p_i，第 i 种结果出现的次数为 n_i。如果用随机变量 $X=(X_1,X_2,...,X_k)$ 表示试验所有可能结果的次数，其中 X_i 表示第 i 种结果出现的次数，那么随机变量 X 服从多项分布。

$$\begin{aligned} P\left(X_1=n_1,\cdots,X_k=n_k\right) &= \frac{n!}{n_1!\cdots n_k!}p_1^{n_1}\cdots p_k^{n_k} \quad ,\sum_{i=1}^k n_i=n \\ &= n!\prod_{i=1}^k \frac{p_i^{n_i}}{n_i!} \quad ,\sum_{i=1}^k n_i=n \end{aligned}$$

当试验的次数 n 为 1 时，多项分布变成类别分布。类别分布表示试验可能出现的 k 种结果的概率。

7.1.3 贝塔分布

贝塔分布是关于连续变量 $x\in[0,1]$ 的概率分布，它由两个参数 $a>0$ 和 $b>0$ 确定:

$$\operatorname{Beta}(x\mid a,b)=\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}x^{a-1}(1-x)^{b-1}=\frac{1}{B(a,b)}x^{a-1}(1-x)^{b-1}$$

其中, $\Gamma(s)$ 为 Gamma 函数:

$$\Gamma(s) = \int_0^{+\infty} t^{s-1} e^{-t} dt$$

$B(a,b)$ 为 Beta 函数，用来做归一化:

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

7.1.4 狄利克雷分布

狄利克雷分布是一种多元连续随机变量的概率分布，是贝塔分布的扩展。在贝叶斯学习中，狄利克雷分布常作为多项分布的先验分布使用。狄利克雷分布是关于一组 d 个连续变量 $x_i \in [0,1]$ 的概率分布，$\sum_i x_i = 1_0$ 。令 $\mu = (\mu_1, \mu_2, \cdots, \mu_d)$ ，参数 $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_d)$ ，其中 $\alpha_i > 0$ 且 $\hat{\alpha} = \sum_i \alpha_i$ 。

$$\mathrm{Dir}(x \mid \alpha) = \frac{\Gamma(\sum_{i=1}^d \alpha_i)}{\prod_{i=1}^d \Gamma(\alpha_i)} \prod_{i=1}^d x_i^{\alpha_i - 1}$$

当 $d = 2$ 时，狄利克雷分布退化为贝塔分布。

7.1.5 共轭先验

贝叶斯学习中常使用共轭分布。如果后验分布与先验分布属于同类，则先验分布与后验分布称为 **共轭分布**，先验分布称为共轭先验。

如果多项分布的先验分布是狄利克雷分布，则其后验分布也为狄利克雷分布，两者构成共轭分布。作为先验分布的狄利克雷分布的参数又称为超参数。使用共轭分布的好处是便于从先验分布计算后验分布。

7.2 MLE/MAP/Bayesian Learning：多分类

1. 二分类 $X \in \{0,1\}$, $P(X=0) = \prod_{k=0}^{1} \theta_k^{I(x=k)}$
2. 多分类 $X \in \{1,...,K\}$, $P(X=x) = \prod_{k=1}^{K} \theta_k^{I(x=k)}$ 查表法

Learning：给定 $D = \{x_i\}_{i=1}^n$ 求 θ

- 有目标 goal：损失函数

  MLE

$$P(D;\theta) = \prod_{i=1}^n \prod_{k=1}^6 \theta_k^{I(x=k)} = \prod_{k=1}^6 \theta_k^{\sum_{i=1}^n I(x=k)}$$

$$lnP(D;\theta) = \prod_{k=1}^6 n_k \ln \theta_k$$

求该数据集出现的概率最大时，θ 的取值

$$\frac{\partial \ln P(D;\theta)}{\partial \theta_k}$$

MAP：$\theta \sim P(\theta \mid \alpha)$

$$\theta \sim P(\theta|D) = P(\theta)P(D|\theta)$$
$$\theta^\star = \arg\max_\theta P(\theta|D)$$

贝叶斯学习

MAP 得到 $P(\theta|D)$

$$\forall x\ P(x|D) = \int P(x,\theta|D)d\theta = \int P(\theta|D)P(x|\theta)d\theta = \int \theta_k P(\theta|D)$$

- 有方法 Algorithm：使损失函数达到最小的算法

Inference: 已知 θ，根据 x，求 y

7.3 Bernoulli Mixture Model

有限混合模型（Finite Mixture Model）是一个可以用来表示在总体分布中含有 K 个子分布的概率模型。子分部可以是各种经典的分布模型，包括高斯模型，伯努利模型，多项式模型等。

首先在 D 维空间中定义由 K 个子分布组成的混合模型通用式:

$$P(\boldsymbol{x} \mid \Theta, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k P\left(\boldsymbol{x} \mid \boldsymbol{\theta}_k\right)$$

其中 π_k 为第 k 个子分布的混合系数，也称权重，且满足 $\sum_{k=1}^{K} \pi_k = 1$ ；$\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$。

$\Omega = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ 为一个包含 N 个样本的数据集，且所有样本满足独立同分布。

对于混合模型，通常采用极大似然法（Maximum likelihood) 来估算参数 Θ ，因此我们建立如下混合模型的似然函数，并希望将其最大化，

$$L(\Omega \mid \Theta, \boldsymbol{\pi}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k P\left(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k\right)$$

为便于计算，将上式转化为对数似然函数：

$$\log L(\Omega \mid \Theta, \boldsymbol{\pi}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k P\left(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k\right)$$

一点预备知识：利用贝叶斯定理推导出后验概率 (posterior probability) $\gamma\left(z_{nk}\right)$:

$$\begin{aligned}
\gamma\left(z_{nk}\right) = P\left(z_k = 1 \mid \boldsymbol{x}_n\right) &= \frac{P\left(\boldsymbol{x}_n \mid z_k = 1\right) P\left(z_k = 1\right)}{\sum_{k=1}^{K} P\left(\boldsymbol{x}_n \mid z_k = 1\right) P\left(z_k = 1\right)} \\
&= \frac{\pi_k P\left(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k\right)}{\sum_{k=1}^{K} \pi_k P\left(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k\right)}
\end{aligned}$$

$\gamma\left(z_{nk}\right)$ 表示在已知观测样本 $\boldsymbol{x}_n$ 的条件下，该样本来自于第 k 个子分布 $(z_k = 1)$ 的概率。

7.3.1 混合模型的应用

文档分类

$$\boldsymbol{X}_{n*d} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix}$$

设每一行，即每个样本代表一个文档；每一列代表一个单词出现的次数

$$\begin{aligned} P(y=k) &= \prod_{k=1}^{K} \theta_k^{I(y=k)} \\ P(x_j|y_i) &= \prod_{i=1}^{K}\prod_{j=1}^{V} \Psi_{vk}^{I(y_i=k\ and\ x_j=v)} \\ P(D;\psi) &= \prod P(x_i,y_i) \\ &= \prod P(y_i)P(x_i|y_i) \\ &= \prod_{i=1}^{n}(\prod_{k=1}^{K}\theta_k^{I(y=k)})(\prod_{i=1}^{K}\prod_{j=1}^{V}\Psi_{vk}^{I(y_i=k\ and\ x_j=v)}) \end{aligned}$$

$$\begin{aligned} logP(D|\psi) &= \sum_{i=1}^{n}(\sum_{k=1}^{K} I(y=k)log\theta_k) + \sum_{i=1}^{n}(\sum_{i=1}^{K}\sum_{j=1}^{V} I(y_i=k\ and\ x_j=v)log\Psi_{vk}) \\ &= \sum_{k=1}^{K} c_k log\theta_k + \sum_{i=1}^{K}\sum_{j=1}^{V} c_{k,v} log\Psi_{vk} \end{aligned}$$

求导即可得 θ_k, ψ_{vk}

....

抛硬币

y	A	A	B	B	B	A	A	A	B	B
x	1	0	0	1	1	0	0	1	0	1

$$P(A|黑) = \frac{P(A)P(黑|A)}{P(黑)} = \frac{P(A)P(黑|A)}{P(A)P(黑|A)+P(B)P(黑|B)}$$

7.3.2 二维伯努利分布的混合模型

二维伯努利分布是关于二维布尔向量 $\boldsymbol{x}=[x_1,x_2]$ 的概率分布，其中 $x_1,x_2 \in \{0,1\}$ ，且满足 $x_1+x_2=1$ ，即 x_1,x_2 中只有一个为 1 ；设参数向量 $\boldsymbol{\theta}=[\theta_1,\theta_2]$ ， $\theta_1,\theta_2 \in [0,1]$ ，分别表示 $x_1=1$ 以及 $x_2=1$ 的概率，且满足 $\theta_1+\theta_2=1$ 。其概率分布函数为:

$$\begin{aligned} P(\boldsymbol{x} \mid \boldsymbol{\theta}) &= \theta_1^{x_1}\theta_2^{x_2} \\ &= \theta_1^{x_1}\left(1-\theta_1\right)^{1-x_1} \end{aligned}$$

由此可见，二维伯努利就是我们所熟知的伯努利分布，这里称其为二维是为了与后面介绍的多维伯努利相一致。二维伯努利最简单的应用就是抛硬币问题，一个硬币共有两个面，每次实验只会出现一个面朝上。

二维伯努利混合模型的对数似然函数如下所示：

$$\log L(\Omega \mid \Theta, \pi)=\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \theta_{k,1}^{x_{n,1}}\left(1-\theta_{k,1}\right)^{1-x_{n,1}} \tag{8}$$

其中，$\theta_{k,1}$ 表示第 k 个子分布中参数向量 $\boldsymbol{\theta}_k$ 的第 1 维参数，$x_{n,1}$ 表示第 n 个样本的第 1 维变量值。然后让 (8) 对参数 $\theta_{k,1}$ 求导并令其导数为 0：

$$\begin{aligned}
\frac{\log L(\Omega \mid \Theta, \boldsymbol{\pi})}{\partial \theta_{k,1}} &=\sum_{n=1}^{N} \frac{\partial}{\partial \theta_{k,1}} \log \sum_{k=1}^{K} \pi_k \theta_{k,1}^{x_{n,1}}\left(1-\theta_{k,1}\right)^{1-x_{n,1}} \\
&=\sum_{n=1}^{N} \frac{\pi_k \theta_{k,1}^{x_{n,1}}\left(1-\theta_{k,1}\right)^{1-x_{n,1}}}{\sum_{k=1}^{K} \pi_k \theta_{k,1}^{x_{n,1}}\left(1-\theta_{k,1}\right)^{1-x_{n,1}}} \frac{\frac{\partial}{\partial \theta_{k,1}} \theta_{k,1}^{x_{n,1}}\left(1-\theta_{k,1}\right)^{1-x_{n,1}}}{\theta_{k,1}^{x_{n,1}}\left(1-\theta_{k,1}\right)^{1-x_{n,1}}} \\
&=\sum_{n=1}^{N} \gamma\left(z_{nk}\right) \frac{\partial}{\partial \theta_{k,1}} \log \left(\theta_{k,1}^{x_{n,1}}\left(1-\theta_{k,1}\right)^{1-x_{n,1}}\right) \\
&=\sum_{n=1}^{N} \gamma\left(z_{nk}\right) \frac{\partial}{\partial \theta_{k,1}}\left(x_{n,1} \log \theta_{k,1}+\left(1-x_{n,1}\right) \log \left(1-\theta_{k,1}\right)\right) \\
&=\sum_{n=1}^{N} \gamma\left(z_{nk}\right)\left(\frac{x_{n,1}}{\theta_{k,1}}-\frac{1-x_{n,1}}{1-\theta_{k,1}}\right) \\
&=0
\end{aligned}$$

由此可得

$$\begin{aligned}
\theta_{k,1} &=\frac{\sum_{n=1}^{N} \gamma\left(z_{nk}\right) x_{n,1}}{\sum_{n=1}^{N} \gamma\left(z_{nk}\right)} \\
\theta_{k,2} &=1-\theta_{k,1}
\end{aligned}$$

7.3.3 混合系数 π_k 的求解

对于混合系数 π_k 的求解，由于 π_k 满足约束 $\sum_{k=1}^{K} \pi_k=1$ ，因此在对数似然函数通用表达式末尾添加添加惩罚项 $\lambda\left(1-\sum_{k=1}^{K} \pi_k\right)$ ，因此得到加了惩罚项的对数似然函数 $p \log L(\Omega \mid \Theta, \pi)$

$$p \log L(\Omega \mid \Theta, \boldsymbol{\pi})=\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k P\left(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k\right)+\lambda\left(1-\sum_{k=1}^{K} \pi_k\right)$$

让上式对 π_k 求导得:

$$\begin{aligned}
\frac{p \log L(\Omega \mid \Theta, \pi)}{\partial \pi_k} &=\sum_{n=1}^{N} \frac{\partial}{\partial \pi_k} \log \sum_{k=1}^{K} \pi_k P\left(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k\right)-\lambda \\
&=\sum_{n=1}^{N} \frac{\gamma\left(z_{nk}\right)}{\pi_k}-\lambda=0
\end{aligned}$$

对上式最后一行两边同乘 π_k ，并对 k 求和得:

$$\sum_{k=1}^{K} \sum_{n=1}^{N} \gamma\left(z_{nk}\right)-\lambda \sum_{k=1}^{K} \pi_k=0$$

因此可得:

$$\lambda = N$$
$$\pi_k = \frac{\sum_{n=1}^{N} \gamma\left(z_{nk}\right)}{N}$$

7.4 EM 算法

EM 算法是一种迭代算法，用于含有隐变量（Hidden variable）的概率模型参数的最大似然估计。该算法流程如下：

1. 确定混合数 K ，并初始化参数。
2. E-step：依据当前参数，计算每个数据 x_n 来自子模型 k 的后验概率 $\gamma(z_{nk})$。
3. M-step：根据不同分布，计算模型参数 Θ 以及混合系数 π 。
4. 重复计算 E-step 和 M-step 直至收敛（$||\Theta_{i+1} - \Theta_i|| < \epsilon$ 是一个很小的正数）。

嘻嘻”-... ....- —-. ——-... ....- —-. ——-... –... .—- ..-..... —.. -.. ..—”

8 Lecture 8

2022.4.11 **Outline**

- Homework
- EM
- EM for xxx

$$\begin{aligned}
\log P(x_i;\theta) &= \log \sum_{z_i} P(x_i, z_i; \theta) \\
&= \log \sum_{k=1}^{K} P(x_i, z_i = k; \theta) \\
&= \log \sum_{k=1}^{K} Q_i(z_i = k) \frac{P(x_i, z_i = k; \theta)}{Q_i(z_i = k)} \\
&= \log \mathbb{E}_{Z_i \quad Q_i(Z_i)} [\frac{P(x_i, Z_i; \theta)}{Q_i(Z_i)}] \\
&\geq \mathbb{E}_{Z_i \sim Q_i(Z_i)} [\log \frac{P(x_i, Z_i; \theta)}{Q_i(Z_i)}]
\end{aligned}$$

Jensen Inequality

$$\boxed{\log \mathbb{E}[X] \geq \mathbb{E}[\log X]}$$

z_i	0	1
$mathbbP$	$1-\pi$	π

x_i	$z_i=0$	0	1
	$mathbbP$	$1-p$	p

8.1 EM 算法

E-step

$$Q_i(\theta;\theta^t)=\mathbb{E}_{Z_i\sim\mathbb{P}(Z_i|x_i;\theta^t)}[\log P(x_i,z_i;\theta)]+H(Z_i)$$

M-step

$$\begin{aligned}\theta^{t+1}&=\arg\max_\theta\sum_{i=1}^n(\sum_{k=1}^K\gamma_{ik}\log P(x_i,z_i=k;\theta))\\&=\arg\max_\theta\sum_{k=1}^K\sum_{i=1}^n\gamma_{ik}\log(P(z_i=k;\theta)P(x_i|z_i=k;\theta))\\&=\arg\max_\theta\sum_{k=1}^K(\sum_{i=1}^n\log(P(z_i=k;\theta)+\sum_{i=1}^n\log(P(x_i,z_i=k;\theta)))\end{aligned}$$

8.2 两个例子

Example 1 (Three coins)

$z_i\in\{0,1\}$

$$Q(\theta;\theta^t)=\sum_{i=1}^n\gamma_{i0}\log(1-\pi)+\sum_{i=1}^n\gamma_{i0}\log p^{x_i}(1-p)^{1-x_i}+\sum_{i=1}^n\gamma_{i1}\log\pi+\sum_{i=1}^n\gamma_{i1}\log q^{x_i}(1-q)^{1-x_i}$$

补充 p,q,π 计算

作业

Example 2. GMM

x_i	$z_i=1$	0	1
	$mathbbP$	$1-q$	q