机器学习算法笔记 v2.0

Xinran Wei

2022 年 8 月 7 日

目录

Part I

准备

1 写在前面

1.1 特点和适用人群

本书是一本机器学习算法理论推导的笔记，**不是**机器学习教科书，**不是**机器学习编程指导书，**不是**机器学习调参指导书，也**不能**替代课程教材和课件。本书不会包括任何编程语言的算法实现，只罗列各类算法的核心步骤及其推导。本书不是入门书籍，适合有一定机器学习基础的读者阅读。

本书**适合的人群**：

- 机器学习工程师：用于算法查阅
- 在校本科生或研究生：用于辅助课程学习
- 研究人员：用于辅助科研
- 教师：用于授课准备

本书**不适合的人群**：

- 机器学习 0 基础，想要入门的小白
- 想要获得工程上有效的调参方法的机器学习工程师
- 想要寻找算法代码实现的代码学习者

1.2 内容安排

本书第2首章先会介绍全书主体部分用到的**数学知识**。回归、分类、降维、聚类是机器学习的四个基本任务或基本问题。这些问题的定义在1.4节中给出。接下来，本书将介绍针对这四个基本任务的非概率机器学习方法：第3章介绍**回归和分类**方法，第4章介绍**降维**方法，第5章介绍**聚类**方法。

接下来，本书将关注基于概率的机器学习方法。这是一个算法大家族，借助概率工具，我们不仅能对很多机器学习算法有更本质的认知，还能建立一些更加强大的算法。第6章介绍一系列通用的概率模型**精确或近似求解**方法。它们构成了这一部分所有章节的基础。第7章介绍一些**基础概率机器学习模型**，包括两个线性回归和 Logistic 回归的概率视角分析。第8章介绍一系列**高斯概率模型**，涵盖了回归、分类、降维和聚类任务。

本书接下来介绍**概率图模型**。这是一类关注变量内部依赖关系的概率模型，而不像前几章介绍的概率模型一样对所有样本的分量同等对待。第9章介绍一些**基础知识**以及精确推断方法，第10章和第11章分别介绍具体的**二值概率图**模型和**时序概率图**模型。

最后，在第12章，我们简单介绍利用深度神经网络工具建立的的**深度生成模型**。尽管涉及神经网络，但本书不包含关于深度神经网络的原理、训练、结构、理论等方面的内容。此外，本书的目的不在于完全地收集各类主流机器学习算法，而在于清晰地讲解部分算法的推导。因此，本书不包含决策树类、集成学习以及基于密度的聚类算法等读者较为熟知的机器学习算法，敬请谅解。

再次强调，本书不包含的内容罗列如下：

- 决策树类算法
- 集成学习算法

- 基于密度的聚类算法
- 深度神经网络的原理、结构、训练、理论
- 各类算法的代码实现

1.3 符号说明

除特殊说明外，本书中的标量一律使用非粗体小写字母，如 λ。向量一律使用粗体小写字母，如 $\mathbf{t}$。矩阵一律使用非粗体大写字母，如 A.

本书中矩阵元素表示为对应矩阵符号的非粗体小写符号，如矩阵 A 第 i 行第 j 列元素 a_{ij}。向量元素表示为对应向量符号的非粗体小写符号，如 $\mathbf{x} = [x_1, x_2, ..., x_n]^T$。如果不特殊说明，本书中所有向量均为列向量，向量右上有转置符号时表示行向量。

除样本集外，当使用向量组成矩阵时，列向量使用带角标的粗体小写字母表示。如 $P = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n] \in R^{m\times n}$。如要表示行向量构成的矩阵，应当写出矩阵转置的组成表示。注意矩阵列向量的元素下标和矩阵元素下标的对应关系：$\mathbf{p}_j = [p_{1j}, p_{2j}, ..., p_{mj}]^T$.

特别地，本书中的（向量）样本集统一表示如下：$X = [\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, \mathbf{x^{(3)}}, ..., \mathbf{x^{(N)}}]^T \in R^{n\times N}$。样本集中的样本索引使用右上角带小括号的数字角标 $\mathbf{x}^{(i)}$。部分算法中，为了表示简便，会要求样本扩充一个常数维度，此时定义扩充后样本为 n 维。

迭代第 t 次的某变量 x 表示为 $x(t)$。

关于**积分**：本书中的积分基本上全部为黎曼积分的定积分形式，且大部分积分变量均为实变量或实向量。为简便起见，在定积分范围为该变量/向量定义的全空间（R^n）时，**省略积分限**。

关于**求导布局**：涉及矩阵的求导时，本书中在不引起歧义的情况下统一使用混合布局，即分子为标量函数时，导数和分母形状一致；分母为标量函数时，导数和分子形状一致。

关于**对数**：本书出现的所有对数均以自然对数 e 为底，书写时以 log 表示，如 $\log f(\mathbf{x})$。

1.4 任务定义

所谓任务就是形式化的问题，是算法解决的目标。本书作为机器学习算法笔记，每一种算法都针对一种具体的任务。在阅读算法时，读者应当首先理清算法针对的是什么任务，解决的是什么问题。

对于机器学习而言，我们有四大**基本任务**：回归、分类、降维、聚类。前两者也被称为有监督学习任务，后者一般为无监督学习任务，但少数降维算法也可以有监督。

回归任务：有样本集 $X = [\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, \mathbf{x^{(3)}}, ..., \mathbf{x^{(N)}}] \in R^{n\times N}$，其中 $\mathbf{x^{(i)}} \in R^n$；以及标签集 $Y = [y^{(1)}, y^{(2)}, .., y^{(n)}]^T$，其中 $\mathbf{y^{(i)}} \in R$。求变换 $f : R^n \to R$，使得 $f(\mathbf{x^{(i)}})$ 尽量好的拟合 $y^{(i)}$。

分类任务：有样本集 $X = [\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, \mathbf{x^{(3)}}, ..., \mathbf{x^{(N)}}] \in R^{n\times N}$，其中 $\mathbf{x^{(i)}} \in R^n$；以及标签集 $Y = [y^{(1)}, y^{(2)}, .., y^{(n)}]^T$，其中 $\mathbf{y^{(i)}} \in L_c = \{c_i | i = 1, ..., c\}$。求变换 $f : R^n \to L_c$，使得 $f(\mathbf{x^{(i)}}) = y^{(i)}$。

降维任务[1]：对数据集 $X = [\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, \mathbf{x^{(3)}}, ..., \mathbf{x^{(N)}}] \in R^{n\times N}$，其中 $\mathbf{x^{(i)}} \in R^n$，求编码和解码变换 $f : R^n \to R^m, g : R^m \to R^n, m < n$，使得，重构样本 $g(f(\mathbf{x^{(i)}}))$ 尽量好的拟合原样本 $x^{(i)}$。

聚类任务：对数据集 $X = [\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, \mathbf{x^{(3)}}, ..., \mathbf{x^{(N)}}] \in R^{n\times N}$，其中 $\mathbf{x^{(i)}} \in R^n$，求变换 $f : R^n \to L_c, L_c = \{c_i | i = 1, ..., c\}$ 使得相似数据尽量被分到同一个类别标签中。

此外，对于各种概率模型，我们有如下概率机器学习任务：**参数估计任务**、**分布估计任务**、**分布采样任务**，详见第6章。对于概率图模型，我们有**分布估计任务**和**参数学习任务**，详见第9.1节。特别地，对于部分含

[1]注：这个定义使用了编码器-解码器的思路。但其实降维只是其中编码器的部分。并非所有降维算法都会显式给出解码器。

有隐变量的概率图，我们还有**模型评估任务**和**隐变量推断任务**，详见第11.2节。这些任务我们统称为**模型相关任务**。

机器学习基本任务和上述模型相关任务之间的关系是：对于特定模型，机器学习基本任务可以转化为模型相关任务。例如：隐马尔可夫模型（HMM）可用于解决序列任务的分类问题，需要针对每类构造并学习一个 HMM。这样，基本任务分类问题就转化成了含隐变量模型的参数学习和评估问题。

1.5 缩写索引表

表1是所有全书中提到的缩写的索引表，以供查阅。索引表按照缩写在正文中出现顺序排序。

表 1: 全书缩写索引表

缩写	**全拼**	**中文含义**
PRML	Pattern Recognition and Machine Learning	模式识别与机器学习
NNDL	Neural Network and Deep Learning	神经网络与深度学习
KKT 条件	Karush-Kuhn-Tucker Condition	(一种约束优化解的条件)
LASSO	Least absolute shrinkage and selection operator	最小绝对值收敛和选择算子
SVM	Support Vector Machine	支持向量机
RBF	Radial Basis Function	径向基函数
SVR	Support Vector Regression	支持向量回归
PCA	Principal Component Analysis	主成分分析
LPP	Locality Preserving Projection	局部保持投影
MDS	Multiple Dimensional Scaling	多维度缩放
ISOMAP	Isometric Mapping	等距特征映射
LLE	Lacally Linear Embedding	局部线性嵌入
LE	Laplacian Eigenmap	拉普拉斯特征图
SOM	Self-Organizing Map	自组织映射
K-L 散度	Kullback-Leibler Divergence	(一种概率分布差异的度量)
MAP	Maximum A Posteriori	最大后验
MLE	Maximum Likelihood Estimation	最大似然估计
E-M 算法	Expectation-Maximization Algorithm	期望-最大化算法
ELBO	Evidence Lower BOund	证据下界
VI	Variational Inference	变分推断
CAVI	Coordinate Ascent Variational Inference	坐标下降变分推断
SGVI	Stochastic Gradient Variational Inference	随机梯度变分推断
MCMC	Markov Chain Monte Carlo	马尔可夫链蒙特卡洛
M-H 采样	Metropolis-Hastings Sampling	(一种采样方法)
BFGS 算法	BFGS Algorithm	(一种拟牛顿优化算法)
GDA	Gaussian Discriminant Analysis	高斯判别分析
GPR	Gaussian Process Regression	高斯过程回归
GMM	Gaussian Mixture Model	高斯混合模型

全书缩写索引表 – 接上页

缩写	全拼	中文含义
FA	Factor Analysis	因子分析
MRF	Markov Random Field	马尔科夫随机场
SBN	Sigmoid Belief Network	Sigmoid 信念网络
BM	Boltzmann Machine	玻尔兹曼机
RBM	Restricted Boltzmann Machine	受限玻尔兹曼机
DBM	Deep Boltzmann Machine	深度玻尔兹曼机
DBN	Deep Belief Network	深度信念网络
HMM	Hidden Markov Model	隐马尔可夫模型
CRF	Conditional Random Field	条件随机场
VAE	Variational Auto Encoder	变分自动编码器

1.6 算法索引表

为便于读者使用，我们将全书介绍的所有算法总结为表2（非概率部分）和表3（概率部分）。表中的算法按照出现先后顺序排序。我们不仅列出了算法的名字，还提供了到正文算法介绍的链接，以及算法解决的任务类型和解的类型。读者可以借助此表快速查阅相关算法。

表 2: 算法索引表 - 非概率方法

算法	位置	任务	解类型
线性回归	3.1.1	回归	闭式解
岭回归	3.1.2	回归	闭式解
LASSO 回归	3.1.2	回归	迭代解
多项式回归	3.1.3	回归	闭式解
Logistic 回归	3.1.4	分类	迭代解
Softmax 回归	3.1.4	分类	迭代解
支持向量机	3.2.1	分类	迭代解
核支持向量机	3.2.2	分类	迭代解
软间隔核支持向量机	3.2.3	分类	迭代解
非线性最小二乘回归	3.3.1	回归	闭式解
k 近邻回归	3.3.2	回归	确定解
核回归	3.3.3	回归	闭式解
支持向量回归	3.3.4	回归	迭代解
Fisher 线性判别	3.4.1	分类	闭式解
k 近邻分类	3.4.2	分类	确定解
主成分分析	4.1.1	降维	闭式解
局部保持投影	4.1.2	降维	闭式解

算法索引表 - 非概率方法 – 接上页

算法	位置	任务	解类型
核主成分分析	4.2.1	降维	闭式解
多维度缩放	4.2.2	降维	闭式解
等距特征映射	4.2.3	降维	确定解
局部线性嵌入	4.2.4	降维	闭式解
拉普拉斯特征图	4.2.5	降维	闭式解
k 均值聚类	5.1.1	聚类	迭代解
模糊 k 均值聚类	5.1.2	聚类	迭代解
核 k 均值聚类	5.1.3	聚类	迭代解
谱聚类	5.2	聚类	闭式解
自组织映射	5.3	聚类	迭代解

表 3: 算法索引表 - 概率方法

算法	位置	任务	解类型	模型类型
E-M 算法	6.2	参数点估计	迭代解	——
坐标下降变分推断	6.3.1	分布估计	迭代解	——
随机梯度变分推断	6.3.1	分布估计	迭代解	——
拉普拉斯近似	6.3.4	分布估计	近似解	——
直接采样	6.4.1	分布采样	——	——
接受-拒绝采样	6.4.2	分布采样	——	——
MCMC 采样	6.4.3	分布采样	——	——
M-H 采样	6.4.4	分布采样	——	——
Gibbs 采样	6.4.5	分布采样	——	——
重要性采样	6.4.6	积分采样	——	——
朴素贝叶斯	7.1	分类-参数估计	闭式解	判别式
线性回归-最大似然	7.2.1	回归-参数估计	闭式解	判别式
线性回归-最大后验	7.2.2	回归-参数估计	闭式解	判别式
线性回归-后验分布	7.2.3	回归-分布估计	闭式解	判别式
线性回归-预测模型	7.2.4	回归-分布估计	闭式解	判别式
线性回归-超参数证据近似	7.2.5	回归-超参数估计	迭代解	判别式
线性回归-超参数 EM 算法	7.2.6	回归-超参数估计	迭代解	判别式
Logistic 回归-最大后验	7.3.1	分类-参数估计	迭代解	判别式
Logistic 回归-Laplacian 近似	7.3.2	分类-分布估计	近似解	判别式
最大熵-梯度上升法	7.4.1	分类-参数估计	迭代解	判别式
最大熵-改进迭代尺度法	7.4.2	分类-参数估计	迭代解	判别式
高斯判别分析	8.1	分类-参数估计	闭式解	生成式

算法索引表 - 概率方法 – 接上页

算法	位置	任务	解类型	模型类型
高斯过程回归	8.2.1	回归-分布估计	闭式解	判别式
高斯混合模型-EM 算法	8.3.1	聚类-参数估计	迭代解	生成式
因子分析-EM 算法	8.4.2	降维-点估计	迭代解	生成式
Sigmoid 信念网络-醒眠算法	10.1.2	参数估计	迭代解	生成式
玻尔兹曼机-梯度上升	10.2.1	参数估计	迭代解	生成式
玻尔兹曼机-变分推断	10.2.2	分布估计	迭代解	生成式
受限玻尔兹曼机-梯度上升	10.3.1	参数估计	迭代解	生成式
受限玻尔兹曼机-对比散度	10.3.2	参数估计	迭代解	生成式
隐马尔可夫-前向算法	11.2.3	模型评估	闭式解	生成式
隐马尔可夫-后向算法	11.2.3	模型评估	闭式解	生成式
隐马尔可夫-Viterbi 算法	11.2.4	隐变量推断	闭式解	生成式
隐马尔可夫-Baum-Welch 算法	11.2.5	参数估计	迭代解	生成式
条件随机场-Viterbi 算法	11.3.3	隐变量推断	闭式解	判别式
条件随机场-改进迭代尺度法	11.3.4	参数估计	迭代解	判别式
变分自编码器	12.1	参数估计	迭代解	生成式

1.7 致谢

感谢 D 同学、L 同学在本书写作过程中的支持和帮助。感谢 C 同学精神上的支持与鼓励。感谢 Ian Goodfellow 教授的 Deep Learning（也被称为“花书”），Christopher Bishop 教授的 Pattern Recognition and Machine Learning (PRML)，邱锡鹏教授的《神经网络与深度学习》(NNDL) 以及李航博士的《统计机器学习》等机器学习经典教材，它们在本书的写作中起到了很大作用。

在这里，特别鸣谢 bilibili UP 主 @shuhuai008 。他的机器学习白板推导系列视频为本书我厘清很多基本概念提供了相当大的帮助。

1.8 版权声明

1.9 联系作者

读者阅读本书过程中，若发现任何排版、公式、推导、文字等方面的错误，或有任何对本书的建议，请发送邮件至作者邮箱 weixr0605@sina.com，诚挚感谢您的帮助！

2 数学知识

本书中用到了很多数学知识和结论，统一在这里列举。由于本书不是教科书，不追求数学体系的完整，只列出使用的概念、部分结论和证明。每个结论都标注了用到该部分知识的位置。

2.1 线性代数

欧氏距离（第3.2节）
欧氏空间 R^n 中，向量 $\mathbf{y}$ 到超平面 $\mathbf{w}^T\mathbf{x}+b=0$ 的欧氏距离为

$$d(\mathbf{y})=\frac{|\mathbf{w}^T\mathbf{y}+b|}{||w||} \tag{2.1}$$

相似对角化（多处用到，非常常用）
实对称矩阵一定可以相似对角化为特征值组成的对角阵。若 $Q^T=Q\in R^{n\times n}$，则

$$Q=P^T\Lambda P,\ P^TP=I \tag{2.2}$$

如果矩阵有正定/负定/半正定/半负定性质，其特征值也会有相应的性质。例如，若有 $Q\geq 0$，则有 $\Lambda\geq 0$。
分块矩阵行列式（第8.2，8.4节）
对于分块矩阵

$$X=\begin{bmatrix}A & B\\ C & D\end{bmatrix}$$

若 A 可逆，则有

$$|X|=|A||D-CA^{-1}B| \tag{2.3}$$

2.1.1 分块矩阵求逆

分块矩阵的求逆公式在第8.2节、8.4节等处都有用到。假设有分块矩阵

$$X=\begin{bmatrix}A & B\\ C & D\end{bmatrix}$$

先列出结论：若 $D-CA^{-1}B$ 可逆，则 X 的逆矩阵为

$$\begin{bmatrix}A & B\\ C & D\end{bmatrix}^{-1}=\begin{bmatrix}A^{-1}+A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1}\\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1}\end{bmatrix} \tag{2.4}$$

证明. 证明过程采用高斯消元法。

$$\begin{bmatrix}I & 0\\ 0 & I\end{bmatrix}\begin{bmatrix}A & B\\ C & D\end{bmatrix}\begin{bmatrix}I & 0\\ 0 & I\end{bmatrix}=\begin{bmatrix}A & B\\ C & D\end{bmatrix}$$

首先将左上角化为单位阵

$$\begin{bmatrix} A^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & A^{-1}B \\ C & D \end{bmatrix}$$

第二行减掉第一行左乘 CA^{-1}

$$\begin{bmatrix} A^{-1} & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & A^{-1}B \\ 0 & D - CA^{-1}B \end{bmatrix}$$

若右下角可逆，第二列右乘右下角的逆矩阵，记为 $Y^{-1} = (D - CA^{-1}B)^{-1}$

$$\begin{bmatrix} A^{-1} & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & Y^{-1} \end{bmatrix} = \begin{bmatrix} I & A^{-1}BY^{-1} \\ 0 & I \end{bmatrix}$$

接下来只需第二列减掉第一列右乘右上角即可

$$\begin{bmatrix} A^{-1} & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & -A^{-1}BY^{-1} \\ 0 & Y^{-1} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

于是我们现在得到了形如 $PXQ = I$ 的式子。两边左乘 Q，有 $QPXQ = Q$。Q 是下三角阵，一定可逆，所以有 $QPX = I$。根据逆矩阵定义，有 $X^{-1} = QP$，即

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} I & -A^{-1}BY^{-1} \\ 0 & Y^{-1} \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ -CA^{-1} & I \end{bmatrix}$$

即

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BY^{-1}CA^{-1} & -A^{-1}BY^{-1} \\ -Y^{-1}CA^{-1} & Y^{-1} \end{bmatrix}$$

完全展开的形式为

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

同理，若 $A - BD^{-1}C$ 可逆，逆矩阵也可以写为

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{bmatrix} \tag{2.5}$$

□

2.1.2 矩阵求导公式

矩阵求导公式广泛应用于各类机器学习方法。首先是矩阵求导的全微分法则：若

$$\mathbf{d}f(X) = \mathrm{tr}(A(\mathbf{d}X^T)) \tag{2.6}$$

则有

$$\frac{\partial f}{\partial X} = A \tag{2.7}$$

接下来罗列一些常用求导公式。已知：$A \in R^{N\times N}, B \in R^{n\times n}, X \in R^{n\times N}, u, v \in R^N, P, Q \in R^{N\times N}$，则有

$$\frac{\partial u^T A v}{\partial A} = uv^T \tag{2.8}$$

$$\frac{\partial u^T X^T X v}{\partial X} = X(uv^T + vu^T) \tag{2.9}$$

$$\frac{\partial tr(X^T B X)}{\partial X} = 2BX \tag{2.10}$$

$$\frac{\partial tr(X^T X A)}{\partial X} = 2XA \tag{2.11}$$

$$\frac{\partial tr(A^T X^T X A)}{\partial X} = 2XAA^T \tag{2.12}$$

$$\frac{\partial |A|}{\partial A} = A^\star = |A|A^{-1}{}^2 \tag{2.13}$$

$$\frac{\partial u^T A^{-1} v}{\partial A} = A^{-1} u v^T A^{-1} \tag{2.14}$$

证明. 可根据矩阵全微分进行计算

$$\begin{aligned}\mathbf{d}(u^T A^{-1} v) &= \mathrm{tr}(uv^T \mathbf{d}(A^{-1})) \\ &= \mathrm{tr}(uv^T A^{-1} \mathbf{d}(A) A^{-1}) \\ &= \mathrm{tr}(A^{-1} uv^T A^{-1} \mathbf{d}(A))\end{aligned}$$

□

$$\frac{\partial \,\mathrm{tr}(APA^T Q)}{\partial A} = (Q^T + Q)AP \tag{2.15}$$

证明.

$$\begin{aligned}\mathbf{d}\,\mathrm{tr}(APA^T Q) &= \mathrm{tr}(\mathbf{d}(A)PA^T Q + A\mathbf{d}(PA^T Q)) \\ &= \mathrm{tr}(PA^T Q\mathbf{d}(A)) + \mathrm{tr}(QAP\mathbf{d}(A^T)) \\ &= \mathrm{tr}(\mathbf{d}(A^T)Q^T AP) + \mathrm{tr}(QAP\mathbf{d}(A^T)) \\ &= \mathrm{tr}(Q^T AP\mathbf{d}(A^T)) + \mathrm{tr}(QAP\mathbf{d}(A^T)) \\ &= \mathrm{tr}((Q^T + Q)AP\mathbf{d}(A^T))\end{aligned}$$

□

2.2 最优化理论

机器学习模型的参数学习往往是一个最优化问题的求解过程，需要使用最优化理论的工具。例如，本书中的很多算法求解（太多了，不一一列举）都是等式约束的最优化问题，需要使用 Lagrange 乘子法。在介绍支持向量机（第3.2节）和最大熵模型（第7.4）时，也用到了本节介绍的对偶理论等工具。

[2]后面一个等号成立的条件是 A 可逆

2.2.1 基本概念

最优化理论是应用数学的一大重要分支。简单来说，最优化理论研究的是以下这一类问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) = 0,\ i = 1, 2, ..., k \\ & h_j(\mathbf{x}) \leq 0,\ j = 1, 2, ..., l \end{aligned} \tag{2.16}$$

在这一问题中，$\mathbf{x}$ 称为**决策变量**，简称变量。决策变量可以是连续变量，也可以是离散变量。本书中我们只考虑连续变量。函数 $f(\mathbf{x})$ 称为**目标函数**。$g_i(\mathbf{x}) = 0$ 称为等式约束，$h_j(\mathbf{x}) \leq 0$ 称为不等式约束。如果没有任何约束，优化问题称为**无约束优化问题**，反之则为**约束最优化问题**。若 f, g, h 函数均为线性函数，该问题称为**线性规划问题**，否则称为**非线性规划**问题。

对于约束优化问题，满足约束条件的所有 $\mathbf{x}$ 构成 $\mathbf{x}$ 的定义域的一个子集，称为**可行域**。若可行域为空集，称优化问题**不可行**。可行域内的 $\mathbf{x}$ 的取值称为**可行解**。可行解中使目标函数最小的解称为最优解。如果目标函数可能取无穷小，称优化问题**无界**，反之称为**有界**。对于有界可行的最优化问题，最优解可能有 0 个、1 个、有限多个或无限多个。

对于一个不等式约束 $h_j(\mathbf{x}) \leq 0$，若 $\mathbf{x}$ 的取值使其取等，说明 $\mathbf{x}$ 处于可行域的由约束 $h_j(\mathbf{x}) \leq 0$ 定义的边界上，此时称该约束为**起作用约束**。

若函数 f 为凸函数，可行域 D 为凸集，称问题为**凸优化问题**。线性规划问题是一类特殊的凸优化问题。凸优化问题有一些很好的性质，后面小节会提到。本书中的优化问题大部分都是凸优化问题。

式2.16中的目标函数为最小化形式。如果需要求解最大化问题，可以通过加负号转化为最小化问题。同时，式2.16中的不等式约束只写出了小于等于形式。如果有小于、大于或不等于形式的约束，可以通过一定的方式转换为小于等于形式的约束。

2.2.2 Lagrange 条件

对于无约束非线性规划问题，已经有很多较为成熟的迭代求解算法，如梯度下降、坐标下降、牛顿法、拟牛顿法等等，本书不再赘述。对于约束最优化问题，可以通过一定的方式找到最优解满足的必要条件，然后求解较为简单的必要条件。本节重点关注这种转换方法。

具体来说，考虑约束问题可行域中的点 $\mathbf{x}$。若向某方向移动一微小距离后仍在可行域中，称该方向 $\mathbf{p}$ 为**可行方向**。假设目标函数和约束条件均连续可微，那么可以得到这样的感性结论：**在最优解处，目标函数梯度方向不能是可行方向**。

首先考虑较为简单的情形，即一个凸的约束优化问题只有等式约束。可以想象，在决策变量所在的 n 维欧氏空间中，一个等式约束 $g_i(\mathbf{x}) = 0$ 定义了一个 $n-1$ 维超平面。在任何可行解处，可行方向 $\mathbf{p}$ 在超平面内，也就是可行方向垂直于约束梯度

$$\mathbf{p} \perp \nabla g_i(\mathbf{x})$$

对于 k 个约束而言，这些约束共同作用，取交集。这样，就要求可行方向垂直于各超平面法向量张成的空间

$$\mathbf{p} \perp \sum_{i=1}^{k} \lambda_i \nabla g_i(\mathbf{x}), \forall \lambda$$

现要求“最优解处，目标函数梯度方向不能是可行方向”，也就是说目标函数必须在可行方向空间的正交补空间中，而这正是各超平面法向量张成的空间。也就是说，目标函数的梯度恰为等式约束梯度的线性组合

$$\nabla f(\mathbf{x}) + \sum_{i=1}^{k} \lambda_i \nabla g_i(\mathbf{x}), \forall \lambda = 0$$

上面的结论可以写成等价的较为规范的形式。首先定义只有等式约束的最优化问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) = 0,\ i = 1, 2, ..., k \end{aligned} \tag{2.17}$$

若该问题存在最优解 $\mathbf{x}^\star$，则 $\mathbf{x}^\star$ 一定也满足

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda) = 0 \\ \frac{\partial}{\partial \lambda} \mathcal{L}(\mathbf{x}, \lambda) = 0 \end{aligned} \tag{2.18}$$

其中

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{k} \lambda_i g_i(\mathbf{x}) \tag{2.19}$$

式2.19称为 Lagrange 函数。式2.18称为 Lagrange 条件。系数 λ 称为 Lagrange 乘子。将约束最优化问题转换为 Lagrange 条件进行求解的方法称为 Lagrange 乘子法。

对于**凸优化问题**，Lagrange 条件是一个最优解存在前提下满足的必要条件。对于非凸情况，在 f 和 g 满足一定的规范性条件的情况下，Lagrange 条件也是最优解必要条件。严谨的证明可以查阅高等数学教材。

2.2.3 KKT 条件

对于不等式约束，我们分两种情况来讨论。如果可行解 $\mathbf{x}$ 处约束条件 $h_j(\mathbf{x}) \leq 0$ 不是起作用约束，那么可行方向和该条件无关。如果可行解 $\mathbf{x}$ 处约束条件 $h_j(\mathbf{x}) \leq 0$ 是起作用约束，那么可行方向不能是 h_j 减小的方向，但可以是 h_j 增加的方向。

因此，对于含有不等式约束的优化问题2.16，最优解处每一个起作用约束梯度和可行方向的内积都要为正。也就是说，目标函数的梯度需要在起作用约束梯度负方向的线性组合半空间内。即

$$\nabla f(\mathbf{x}) = \sum_{i=1}^{k} u_i \nabla g_i(\mathbf{x}) - \sum_{j=1^l} v_j \nabla h_j(\mathbf{x}), \forall \mathbf{u}, \mathbf{v} \geq 0$$

使用比较规范的语言等价书写，我们可以得到下列必要条件，称为 KKT 条件。

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = 0 \\ \frac{\partial}{\partial \mathbf{u}} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = 0 \\ \frac{\partial}{\partial \mathbf{v}} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}) \leq 0 \\ v_j \frac{\partial}{\partial v_j} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = 0, j = 1, 2, ..., l \\ v_j \geq\ 0, j = 1, 2, ..., l \end{aligned} \tag{2.20}$$

其中

$$\mathcal{L}(\mathbf{x},\mathbf{u},\mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^{k} u_i g_i(\mathbf{x}) + \sum_{j=1}^{l} v_j h_j(\mathbf{x}) \tag{2.21}$$

式2.20中的第四行称为互补松弛条件。它的意思是，若第 j 个不等式约束为起作用约束，则有 $h_j = 0$，此时 v_j 可以为正，可行方向不能为 ∇h_j 的方向。若第 j 个不等式约束不起作用，则 $h_j < 0$，v_j 必须为 0，可行方向和 ∇h_j 无关。

KKT 条件对于凸问题是必要条件，满足 KKT 的解不一定是最优解。对于非凸问题，如果问题中的 $f, \mathbf{g}, \mathbf{h}$ 满足约束线性无关条件（Linear Independence Constraint Quality, LICQ），KKT 条件也是最优解的必要条件。这些证明，读者可以查阅相应的最优化教材，本书不再赘述。

2.2.4 Lagrange 对偶理论

在给定 $\mathbf{x}$ 的条件下，Lagrange 函数2.21可以看作是 $\mathbf{u}, \mathbf{v}$ 的函数。考虑 Lagrange 函数对 $\mathbf{u}, \mathbf{v}$ 求最大的优化问题

$$\begin{aligned} &\max_{u,v} \mathcal{L}(\mathbf{x},\mathbf{u},\mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^{k} u_i g_i(\mathbf{x}) + \sum_{j=1}^{l} v_j h_j(\mathbf{x}) \\ &\text{s.t. } v_j \geq 0, j = 1, 2, ..., l \end{aligned} \tag{2.22}$$

对于该问题，我们根据 $\mathbf{x}$ 可行和非可行的不同情况进行讨论。若 $\mathbf{x}$ 在非可行域，即不满足某一约束条件。不妨假设在 $\mathbf{x}$ 处不满足等式约束 $g_i(\mathbf{x}) = 0$，此时调整 $\mathbf{u}$ 对 $L(\mathbf{x},\mathbf{u},\mathbf{v})$ 取最大。若 $g_i(\mathbf{x})$ 为正，则可取 $u_i = +\infty$，反之 $g_i(\mathbf{x})$ 为负，则可取 $u_i = -\infty$。总之，可以使 $L(\mathbf{x},\mathbf{u},\mathbf{v}) = +\infty$，即该优化问题无界。

同理，如果不满足的是不等式约束条件 $h_j \leq 0$，说明 $h_j > 0$。取 $v_j = +\infty$，也可以使 $L(\mathbf{x},\mathbf{u},\mathbf{v}) = +\infty$。

另一方面，如果 $\mathbf{x}$ 在可行域，那么所有 g_i 均为 0，所有 h_i 均非正。此时，u_i 无论如何取值都不会影响 $L(\mathbf{x},\mathbf{u},\mathbf{v})$，$u_i g_i(\mathbf{x})$ 恒为 0。v_i 取任何正值都只可能令 $L(\mathbf{x},\mathbf{u},\mathbf{v})$ 更小，为了最大化只能取 $\mathbf{v} = 0$。因此，$\mathbf{x}$ 在可行域的条件下，优化问题2.22的最优解为

$$\min_{u,v} \mathcal{L}(\mathbf{x},\mathbf{u},\mathbf{v}) = f(\mathbf{x})$$

下面，我们将优化问题2.22的最优解看成 $\mathbf{x}$ 的函数，即

$$m(\mathbf{x}) = \max_{u,v} \mathcal{L}(\mathbf{x},\mathbf{u},\mathbf{v})$$

根据上面的讨论，按照 $\mathbf{x}$ 是否在可行域，$m(\mathbf{x})$ 的取值分为两种情况

$$m(\mathbf{x}) = \begin{cases} +\infty & \mathbf{x} \in C \\ f(\mathbf{x}) & \mathbf{x} \notin C \end{cases}$$

其中 C 表示原问题2.16的可行域。当 $\mathbf{x}$ 在可行域时 $m(\mathbf{x}) = f(\mathbf{x})$，不在可行域时 $m(\mathbf{x})$ 为正无穷。这样，原来的约束最优化问题2.16就和最小化 $m(\mathbf{x})$ 的无约束优化问题等价

$$\min_{\mathbf{x}} m(\mathbf{x})$$

展开定义，我们有

$$\min_{\mathbf{x}} \max_{u,v} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^{k} u_i g_i(\mathbf{x}) + \sum_{j=1}^{l} v_j h_j(\mathbf{x}) \tag{2.23}$$
$$\text{s.t. } v_j \geq 0, j = 1, 2, ..., l$$

问题2.23称为**极小极大问题**，它和原问题是完全等价的。

再定义一个关于 $\mathbf{u}, \mathbf{v}$ 的函数

$$n(\mathbf{u}, \mathbf{v}) = \min_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{v})$$
$$\text{s.t. } \mathbf{v} \geq 0$$

这个函数是 $L(\mathbf{x}, \mathbf{u}, \mathbf{v})$ 关于 $\mathbf{x}$ 的下界，即

$$n(\mathbf{u}, \mathbf{v}) \leq \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}), \forall \mathbf{x}, \mathbf{u}, \mathbf{v} \geq 0$$

又由于 $m(\mathbf{x})$ 无论在 $\mathbf{x}$ 属于可行域还是不属于可行域时都不会小于 $f(\mathbf{x})$，有

$$m(\mathbf{x}) \geq \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}), \forall \mathbf{x}, \mathbf{u}, \mathbf{v} \geq 0$$

综合上述两式，我们有

$$m(\mathbf{x}) \geq n(\mathbf{u}, \mathbf{v}), \forall \mathbf{x}, \mathbf{u}, \mathbf{v} \geq 0 \tag{2.24}$$

考虑对函数 $n(\mathbf{u}, \mathbf{v})$ 求最大的最优化问题，即

$$\max_{\mathbf{u},\mathbf{v}} \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^{k} u_i g_i(\mathbf{x}) + \sum_{j=1}^{l} v_j h_j(\mathbf{x}) \tag{2.25}$$
$$\text{s.t. } \mathbf{v} \geq 0$$

则该问题的最优解一定是极小极大问题最优解的下界，即

$$\max_{\mathbf{u},\mathbf{v}} \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}) \leq \min_{\mathbf{x}} \max_{u,v} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}), \forall \mathbf{x}, \mathbf{u}, \mathbf{v} \geq 0 \tag{2.26}$$

证明.

$$\max_{\mathbf{u},\mathbf{v}} \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = \max_{\mathbf{u},\mathbf{v}} n(\mathbf{u}, \mathbf{v})$$
$$\leq \min_{\mathbf{x}} m(\mathbf{x}) = \min_{\mathbf{x}} \max_{u,v} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v})$$

其中 $\max_{\mathbf{u},\mathbf{v}} n(\mathbf{u}, \mathbf{v}) \leq \min_{\mathbf{x}} m(\mathbf{x})$ 来自于式2.24。 □

问题2.25称为原问题的对偶问题，也称为原问题的极大极小问题。这个问题的内层对于 $\mathbf{x}$ 是一个无约束最优化问题，求解起来比约束最优化问题容易很多。而外层对于 $\mathbf{u}, \mathbf{v}$，这是一个凸问题，而且是一个线性规划问题，求解也十分容易。

式2.26称为弱对偶原理。它对于任何原问题都适用。除此之外，可以证明，当原问题满足一定的条件（凸问题和 Slater 条件）时，对偶问题的最优解就等于极大极小问题最优解，也就是原问题的最优解

$$\max_{\mathbf{u},\mathbf{v}} n(\mathbf{u}, \mathbf{v}) = \min_{\mathbf{x}} m(\mathbf{x}) = f(\mathbf{x}^\star) \tag{2.27}$$

这一原理也称为强对偶原理。其证明因篇幅所限不再介绍，读者可以查阅相关最优化教材。

2.3 概率论

2.3.1 基本概念

概率论中有很多基本概念会在概率机器学习算法中反复用到，在此仅罗列出来。本书不是数学教材，读者如果对这些概念不够熟悉，应当参考专业的概率论教材进行复习。

- 随机变量和概率：离散随机变量、概率质量函数、连续随机变量、概率密度函数
- 概率运算：互斥事件、加和规则、条件概率、乘积规则
- 多元分布：随机向量、联合分布、边缘分布、条件分布、条件独立
- 贝叶斯定理：贝叶斯公式、先验概率、后验概率、似然函数
- 统计量：期望、方差、协方差、条件期望
- 高斯分布：一元高斯分布、多元高斯分布、乘积性质、边缘性质
- 信息论：信息熵、条件熵、交叉熵、KL 散度
- 指数族分布：伯努利分布、二项分布、泊松分布、高斯分布、Beta 分布、Gamma 分布
- 马尔可夫过程：马尔科夫链、概率转移矩阵、n 步概率转移矩阵、常返状态、非周期性、连通性、**平稳分布**

2.3.2 重要概念和结论

条件分布

条件分布 $p(\mathbf{x}|\mathbf{z})$ 的形式本质是：随机变量 $\mathbf{x}$ 的概率密度函数也是 $\mathbf{z}$ 的函数。或者说，同时和 $\mathbf{x},\mathbf{z}$ 有关的一个函数 $f(\mathbf{x},\mathbf{z})$ 遵循随机变量 $\mathbf{x}$ 分布的约束

$$\int f(\mathbf{x},\mathbf{z})d\mathbf{x} = 1, \forall \mathbf{z} \tag{2.28}$$

$$f(\mathbf{x},\mathbf{z}) \geq 0, \forall \mathbf{x},\mathbf{z} \tag{2.29}$$

很常见的一类条件分布是：$\mathbf{x}$ 服从某一简单分布，而该简单分布的参数由变量 $\mathbf{z}$ 控制。例如

$$p(\mathbf{x}|\mathbf{z}) = N(\mathbf{x};\mu(\mathbf{z}),\Sigma(\mathbf{z}))$$

条件独立性

若随机变量 Z,Y,Z 满足 $p(X,Y|Z) = p(X|Z)p(Y|Z)$，或 $p(X|Y,Z) = p(X|Z)$，则称 X 和 Y 关于 Z 条件独立。

条件独立性是概率图模型最重要的理论基础。

Jensen 不等式

对于凹函数 $f(\mathbf{x})$ 满足 $\nabla^2 f \leq 0$，设有任意概率分布 $\mathbf{x} \sim p(\mathbf{x})$，则有

$$E_{\mathbf{x}\sim p(\mathbf{x})}[f(\mathbf{x})] \leq f(E_{\mathbf{x}\sim p(\mathbf{x})}[\mathbf{x}]) \tag{2.30}$$

条件熵

条件熵是条件分布 $p(\mathbf{y}|\mathbf{x})$ 的泛函，其定义为

$$H[p(\mathbf{y}|\mathbf{x})] = \left\langle \int -p(\mathbf{y}|\mathbf{x}) \log p(\mathbf{y}|\mathbf{x}) d\mathbf{y} \right\rangle_{p(\mathbf{x})} \tag{2.31}$$

2.3.3 多元高斯分布

由于本书大量使用多元高斯分布，将其相关结论在一节中集中展示。

定义

$$N(\mathbf{x}; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\} \tag{2.32}$$

概率密度对数

$$\log p(\mathbf{x}|\mu, \Sigma) = -\frac{1}{2}(k \log(2\pi) + \log|\Sigma|) - \frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu) \tag{2.33}$$

信息熵

$$H[p(\mathbf{x}|\mu, \Sigma)] = \frac{k}{2}(\log(2\pi)+1) + \frac{1}{2}\log|\Sigma| \tag{2.34}$$

本条证明使用了第 4 小节的式2.41。

条件分布

考虑满足联合高斯分布的变量 $\mathbf{x} \in R^n, \mathbf{z} \in R^m$

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}; \begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{xz}^T & \Sigma_{zz} \end{bmatrix} \right) \tag{2.35}$$

有条件分布

$$\mathbf{z}|\mathbf{x} \sim N(\mathbf{z}; \mu_{z|x}, \Sigma_{z|x}) \tag{2.36}$$

其中

$$\begin{aligned} \mu_{z|x} =& \mu_{\mathbf{z}} + \Sigma_{xz}^T \Sigma_{xx}^{-1} (\mathbf{x} - \mu_x) \\ \Sigma_{z|x} =& \Sigma_{zz} - \Sigma_{xz}^T \Sigma_{xx}^{-1} \Sigma_{xz} \end{aligned} \tag{2.37}$$

相关期望

关于多元高斯分布 $\mathbf{x} \sim N(\mathbf{x}; \mu, \Sigma)$，有如下常见期望结果

$$\langle \mathbf{x} \rangle_{N(\mathbf{x};\mu,\Sigma)} = \mu \tag{2.38}$$

$$\langle \mathbf{x}\mathbf{x}^T \rangle_{N(\mathbf{x};\mu,\Sigma)} = \Sigma + \mu\mu^T \tag{2.39}$$

$$\langle \mathbf{x}^T\mathbf{x} \rangle_{N(\mathbf{x};\mu,\Sigma)} = \mathrm{tr}(\Sigma + \mu\mu^T) \tag{2.40}$$

证明.

$$\begin{aligned}&\left\langle \mathbf{x}^T\mathbf{x}\right\rangle_{N(\mathbf{x};\mu,\Sigma)} = \left\langle \mathrm{tr}(\mathbf{x}^T\mathbf{x})\right\rangle_{N(\mathbf{x};\mu,\Sigma)}\\ =&\left\langle \mathrm{tr}(\mathbf{x}\mathbf{x}^T)\right\rangle_{N(\mathbf{x};\mu,\Sigma)} = \mathrm{tr}\left(\left\langle \mathbf{x}\mathbf{x}^T\right\rangle_{N(\mathbf{x};\mu,\Sigma)}\right) = \mathrm{tr}(\Sigma+\mu\mu^T)\end{aligned}$$

□

$$\left\langle (\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\rangle_{N(\mathbf{x};\mu,\Sigma)} = k \tag{2.41}$$

证明.

$$\begin{aligned}&I = \left\langle (\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\rangle_{N(\mathbf{x};\mu,\Sigma)}\\ =&\int (\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\frac{1}{\sqrt{(2\pi)^k|\Sigma|}}\exp\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\}d\mathbf{x}\end{aligned}$$

由于协方差矩阵是半正定的，其特征值有非负性，假设 Σ 的相似对角化为

$$\Sigma = P^T\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}P$$

因此有

$$(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu) = (\mathbf{x}-\mu)^TP^T\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}P(\mathbf{x}-\mu)$$

假设

$$\mathbf{t}(\mathbf{x}) = \Lambda^{-\frac{1}{2}}P(\mathbf{x}-\mu)$$

则有

$$\mathbf{x}(\mathbf{t}) = P^T\Lambda^{\frac{1}{2}}\mathbf{t}+\mu$$

$$\begin{aligned}&(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\\ =&(P^T\Lambda^{\frac{1}{2}}\mathbf{t})^T\Sigma^{-1}(P^T\Lambda^{\frac{1}{2}}\mathbf{t})\\ =&\mathbf{t}^T\Lambda^{\frac{1}{2}}PP^T\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}PP^T\Lambda^{\frac{1}{2}}\mathbf{t}\\ =&\mathbf{t}^T\mathbf{t}\end{aligned}$$

因此

$$\begin{aligned}&I = \int (\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\frac{1}{\sqrt{(2\pi)^k|\Sigma|}}\exp\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\}d\mathbf{x}\\ =&\frac{1}{\sqrt{|\Sigma|}}\int \mathbf{t}^T\mathbf{t}\frac{1}{\sqrt{(2\pi)^k}}\exp\{-\frac{1}{2}\mathbf{t}^T\mathbf{t}\}d\mathbf{x}\end{aligned}$$

由半正定性，有

$$d\mathbf{x} = \left||\frac{\partial\mathbf{x}}{\partial\mathbf{t}}|\right| d\mathbf{t} = \left||P\Lambda^{\frac{1}{2}}|\right| d\mathbf{t} = \sqrt{|\Sigma|}d\mathbf{t}$$

因此积分为

$$
\begin{aligned}
I =& \frac{1}{\sqrt{|\Sigma|}} \int \mathbf{t}^T\mathbf{t} \frac{1}{\sqrt{(2\pi)^k}} \exp\{-\frac{1}{2}\mathbf{t}^T\mathbf{t}\}\sqrt{|\Sigma|}d\mathbf{t} \\
=& \int \mathbf{t}^T\mathbf{t} \frac{1}{\sqrt{(2\pi)^k}} \exp\{-\frac{1}{2}\mathbf{t}^T\mathbf{t}\}d\mathbf{t} \\
=& \left\langle \mathbf{t}^T\mathbf{t} \right\rangle_{N(\mathbf{t};\mathbf{0};I_k)} \\
=& \mathrm{tr}(I_k + \mathbf{0}\mathbf{0}^T) = k
\end{aligned}
$$

□

2.4 变分法

读者应对泛函的概念、泛函微分和泛函变分的定义十分熟悉。变分法将用于概率方法中的变分近似方法推导。

2.5 图论基础

图论中有很多基本概念会在各种机器学习算法中反复用到，在此仅罗列出来。本书不是数学教材，读者如果对这些概念不够熟悉，应当参考专业的图论教材进行复习。

- 基础概念：节点、边、节点集、边集、有向边、无向边、有向图、无向图
- 量化指标：度、出度、入度、权值、邻接矩阵、度矩阵、拉普拉斯矩阵
- 分割和团：连通图、子图、割、团、最大团
- 基本算法：拓扑排序、Dijkstra 算法

这些知识将用于基于邻接图的降维算法、概率图模型等多个章节。

Part II

非概率方法

3 回归和分类

3.1 线性回归

回归问题和分类问题是监督学习问题里的基础问题。在较为简单的情况下，这两个问题都可以通过线性回归类算法来解决。这些方法还会在第 7章从概率视角重新回顾。

3.1.1 线性模型

算法	**线性回归模型**
算法简述	对于回归问题，假设样本和标签服从线性关系，求出使平方误差最优的线性系数
已知	样本 $X \in R^{n\times N}$（扩充过），标签 $Y \in R^N$
求	线性系数 $\mathbf{w} \in R^n$ 使得 $\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2}\|\|X\mathbf{w}-\mathbf{y}\|\|^2$
解类型	闭式解
闭式解	$\mathbf{w}^\star = (X^TX)^{-1}X^T\mathbf{y}$

闭式解推导：

$$\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{w}^T\mathbf{x}^{(i)} - y^{(i)})^2 = \frac{1}{2}||X\mathbf{w}-\mathbf{y}||^2$$

$$\frac{\partial E}{\partial \mathbf{w}} = X^T(X\mathbf{w}-\mathbf{y}) = 0$$

$$X^TX\mathbf{w} = X^T\mathbf{y}$$

$$\mathbf{w}^\star = (X^TX)^{-1}X^T\mathbf{y} \tag{3.1}$$

3.1.2 岭回归、LASSO 回归

算法	**岭回归**
算法简述	线性回归中，为防止过拟合，对参数向量 $\mathbf{w}$ 的 L2 范数平方进行限制
已知	样本 $X \in R^{n\times N}$（扩充过），标签 $Y \in R^N$
求	线性系数 $\mathbf{w} \in R^n$ 使得 $\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2}\|\|X\mathbf{w}-\mathbf{y}\|\|^2 + \frac{1}{2}\lambda\|\|\mathbf{w}\|\|^2$
解类型	闭式解
闭式解	$\mathbf{w}^\star = (X^TX+\lambda I)^{-1}X^T\mathbf{y}$

闭式解推导：

$$\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2}||X\mathbf{w} - \mathbf{y}||^2 + \frac{1}{2}\lambda||\mathbf{w}||^2$$

$$\frac{\partial E}{\partial \mathbf{w}} = X^T(X\mathbf{w} - \mathbf{y}) + \lambda\mathbf{w} = 0$$

$$(X^TX + \lambda I)\mathbf{w} = X^T\mathbf{y}$$

$$\mathbf{w}^\star = (X^TX + \lambda I)^{-1}X^T\mathbf{y} \tag{3.2}$$

算法	**LASSO 回归**
算法简述	线性回归中，为防止过拟合，对参数向量 $\mathbf{w}$ 的 L1 范数进行限制
已知	样本 $X \in R^{n\times N}$（扩充过），标签 $Y \in R^N$，学习率 α
求	线性系数 $\mathbf{w} \in R^n$ 使得 $\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2}||X\mathbf{w} - \mathbf{y}||^2 + \lambda||\mathbf{w}||_1$
解类型	迭代解
迭代式	$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(\sum_{i=1}^N (\mathbf{w}^T(t)\mathbf{x}^{(i)} - y^{(i)})\mathbf{x}_j^{(i)} + \operatorname{sgn} w_j)$

迭代求解推导：

$$\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2}||X\mathbf{w} - \mathbf{y}||^2 + \lambda||\mathbf{w}||_1$$

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^N (\mathbf{w}^T\mathbf{x}^{(i)} - y^{(i)})\mathbf{x}_j^{(i)} + \operatorname{sgn} w_j$$

一阶迭代式为

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(\sum_{i=1}^N (\mathbf{w}^T(t)\mathbf{x}^{(i)} - y^{(i)})\mathbf{x}_j^{(i)} + \operatorname{sgn}(w_j)) \tag{3.3}$$

3.1.3 多项式回归

算法	**多项式回归**
算法简述	对于回归问题，假设样本维度为 1，且样本和标签服从 m 次多项式关系，求出使平方误差最优的多项式系数
已知	样本 $X \in R^N$（扩充过），标签 $Y \in R^N$
求	线性系数 $\mathbf{w} \in R^{(m+1)}$ 使得 $\min_{\mathbf{w}} E(\mathbf{w}) = \sum_{j=1}^N ||\sum_{i=0}^m w_i {x^{(j)}}^i - \mathbf{y^{(j)}}||^2$
解类型	闭式解
求解方法	只需将 $\hat{\mathbf{x}} = [1, x, x^2, ..., x^m]^T$ 设为新的样本，再使用线性回归模型即可

3.1.4 Logistic 回归、Softmax 回归

线性回归模型同样可用于分类问题。首先考虑二分类问题。

Logistic 回归

设有 Sigmoid 函数

$$\mathrm{sig}(x) = \frac{1}{1+\exp(-x)} \tag{3.4}$$

对其求导有

$$\begin{aligned}
&\frac{d\,\mathrm{sig}}{dx} = \frac{-\exp(-x)(-1)}{(1+\exp(-x))^2} \\
=&\frac{\exp(-x)}{1+\exp(-x)}\frac{1}{1+\exp(-x)} \\
=&\mathrm{sig}(x)(1-\mathrm{sig}(x))
\end{aligned}$$

即

$$\frac{d\,\mathrm{sig}}{dx} = \mathrm{sig}(x)(1-\mathrm{sig}(x)) \tag{3.5}$$

使用交叉熵损失

$$L(P,Q) = \sum_{i=1}^{c} P_i \log Q_i$$

算法	**Logistic 回归**
算法简述	对于二分类问题，假设样本线性可分，求出使交叉熵损失最小的线性系数
已知	样本 $X \in R^{n\times N}$（扩充过），标签 $y^{(i)} \in \{0,1\}$，学习率 α
求	$\mathbf{w} \in R^n$ 使得 $\min_{\mathbf{w}} L(\mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N}(-y^{(i)}\log\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - (1-y^{(i)})\log(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})))$
解类型	迭代解
迭代式	$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\frac{1}{N}\sum_{i=1}^{N}(\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - y^{(i)})\mathbf{x}^{(i)}$

迭代求解推导:
对损失函数求偏导

$$\begin{aligned}
&\frac{\partial}{\partial\mathbf{w}}L(\mathbf{W}) \\
=&\frac{\partial}{\partial\mathbf{w}}\frac{1}{N}\sum_{i=1}^{N}(-y^{(i)}\log\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - (1-y^{(i)})\log(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))) \\
=&\frac{1}{N}\sum_{i=1}^{N}(-y^{(i)}\frac{\partial}{\partial\mathbf{w}}\log\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - (1-y^{(i)})\frac{\partial}{\partial\mathbf{w}}\log(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))) \\
=&\frac{1}{N}\sum_{i=1}^{N}(-\frac{y^{(i)}}{\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})}\frac{\partial}{\partial\mathbf{w}}\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) + \frac{1-y^{(i)}}{1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})}\frac{\partial}{\partial\mathbf{w}}\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})) \\
=&\frac{1}{N}\sum_{i=1}^{N}(-\frac{y^{(i)}}{\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})} + \frac{1-y^{(i)}}{1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})})\frac{\partial}{\partial\mathbf{w}}\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) \\
=&\frac{1}{N}\sum_{i=1}^{N}(\frac{\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - y^{(i)}}{\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))})\,\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))\mathbf{x}^{(i)} \\
=&\frac{1}{N}\sum_{i=1}^{N}(\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - y^{(i)})\mathbf{x}^{(i)}
\end{aligned}$$

因此，梯度下降迭代式为

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha \frac{1}{N} \sum_{i=1}^{N} (\text{sig}(\mathbf{w}^T \mathbf{x}^{(i)}) - y^{(i)}) \mathbf{x}^{(i)} \tag{3.6}$$

Softmax 回归

对于多分类问题，为简单起见，将标签转化为 one-hot 向量 $\mathbf{y}^{(i)} \in \{\mathbf{y} | \mathbf{y}_i \in 0, 1, i = 1, 2, ..., c, ||\mathbf{y}|| = 1\}$

设有 Softmax 函数 $f(x) = [f_1(x), ..., f_c(x)]^T$，其中

$$f_j(\mathbf{x}^{(i)}) = \frac{\exp(\mathbf{w}_j^T \mathbf{x}^{(i)})}{\sum_{k=1}^{c} \exp(\mathbf{w}_k^T \mathbf{x}^{(i)})} \tag{3.7}$$

这个向量值函数的输出对于任何 $\mathbf{x}$ 都是归一化的，也就是说可以认为是一种后验概率

$$p(y = c | \mathbf{x}) = f_c(\mathbf{x}^{(i)}) \tag{3.8}$$

该函数对参数求偏导，有

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}_k} f_j(\mathbf{x}) =& \frac{\partial}{\partial \mathbf{w}_k} \frac{\exp(\mathbf{w}_j^T \mathbf{x})}{\sum_{k=1}^{c} \exp(\mathbf{w}_k^T \mathbf{x})} \\
=& \frac{\exp(\mathbf{w}_j^T \mathbf{x})(\sum_{k=1}^{c} \exp(\mathbf{w}_k^T \mathbf{x}))\mathbf{x} - \exp(\mathbf{w}_j^T \mathbf{x}) \exp(\mathbf{w}_j^T \mathbf{x})\mathbf{x}}{(\sum_{k=1}^{c} \exp(\mathbf{w}_k^T \mathbf{x}))^2} \\
=& \frac{\exp(\mathbf{w}_j^T \mathbf{x})}{\sum_{k=1}^{c} \exp(\mathbf{w}_k^T \mathbf{x})} \mathbf{x} - \left(\frac{\exp(\mathbf{w}_j^T \mathbf{x})}{\sum_{k=1}^{c} \exp(\mathbf{w}_k^T \mathbf{x})} \right)^2 \mathbf{x} \\
=& (f_j(\mathbf{x}))(1 - f_j(\mathbf{x}))\mathbf{x}
\end{aligned}$$

算法	**Softmax 回归**
算法简述	对于分类问题，假设样本线性可分，求出使平方误差最优的线性系数
已知	样本 $X \in R^{n \times N}$（扩充过），标签 $Y = [\mathbf{y^{(1)}}, \mathbf{y^{(2)}}, ..., \mathbf{y^{(N)}}] \in R^{c \times N}$，学习率 α
求	$\mathbf{W} \in R^{n \times c} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_c]$ 使得 $\min_{\mathbf{W}} E(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{c} y_{ji} \log f_j(\mathbf{x}^{(i)})$， 其中 $f(x) = [f_1(x), ..., f_c(x)]^T$
解类型	迭代解
迭代式	$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha \frac{1}{N} \sum_{i=1}^{N} y_{ki}(1 - f_k(\mathbf{x}^{(i)}))\mathbf{x}^{(i)}$

迭代求解推导：

$$
\begin{aligned}
\frac{\partial E}{\partial \mathbf{w}_k} =& \frac{1}{N}\frac{\partial}{\partial \mathbf{w}_k}\sum_{i=1}^{N}\sum_{j=1}^{c} y_{ji}\log f_j(\mathbf{x}^{(i)}) \\
=& \frac{1}{N}\mathbf{y}_k^{(i)}\frac{\partial}{\partial \mathbf{w}_k}\log f_k(\mathbf{x}^{(i)}) \\
=& \frac{1}{N}\sum_{i=1}^{N} y_{ki}\frac{1}{f_k(\mathbf{x}^{(i)})}\frac{\partial}{\partial \mathbf{w}_k} f_k(\mathbf{x}^{(i)}) \\
=& \frac{1}{N}\sum_{i=1}^{N} y_{ki}\frac{1}{f_k(\mathbf{x}^{(i)})} f_k(\mathbf{x}^{(i)})(1-f_k(\mathbf{x}^{(i)}))\mathbf{x}^{(i)} \\
=& \frac{1}{N}\sum_{i=1}^{N} y_{ki}(1-f_k(\mathbf{x}^{(i)}))\mathbf{x}^{(i)}
\end{aligned}
$$

一阶迭代式为

$$
\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha\frac{1}{N}\sum_{i=1}^{N} y_{ki}(1-f_k(\mathbf{x}^{(i)}))\mathbf{x}^{(i)} \tag{3.9}
$$

3.2 支持向量机

3.2.1 线性 SVM

对线性可分的二分类问题，我们可以认为最优线性分类器是**分开两类，且到所有样本最小间隔最大**的超平面。

定义超平面 w, b 到各样本点的最小间隔

$$
\gamma = 2\min_i \frac{|\mathbf{w}^T\mathbf{x}^{(i)} + b|}{||\mathbf{w}||} \tag{3.10}
$$

问题描述为，求 $\mathbf{w} \in R^n$ 使得 $\max_{w,b}\gamma$，且满足 $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) > 0$

这个问题描述可以简化。首先，对 $\mathbf{w}$ 加一系数放缩不影响结果，因此 k 可以加一条限制，使最小间隔的分子部分为 1. 和不等式约束结合，就可以得到下列约束条件

$$
y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, i = 1, ..., N \tag{3.11}
$$

此时优化目标可以化简，最大化目标等价于最小化分母 $\mathbf{w}$ 的模长，也就是最小化二范数

$$
\min_{w,b}\frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{3.12}
$$

这样我们有一个约束优化问题，也称为 **SVM 标准型**

$$
\begin{aligned}
\min_{w,b} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\
\text{s.t.} \quad & y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, i = 1, ..., N
\end{aligned} \tag{3.13}
$$

算法	**支持向量机**
算法简述	线性可分二分类问题中，求解到所有样本最小间隔最大的超平面
已知	样本 $X \in R^{n\times N}$（不扩充），标签 $Y \in 0,1^N$
求	$\mathbf{w} \in R^n, b \in R$ 使得 $\min_{w,b} \frac{1}{2}\mathbf{w}^T\mathbf{w}$ 使得 $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}+b) \geq 1, i = 1, ..., N$
解类型	迭代解
求解方法	在本节第 4 小节一并描述

SVM 只能用于求解二分类问题。对于多分类问题，可以采取以下几种思路将其转换为二分类问题。

- 一对一法：对任意两个类训练一个 SVM，c 个类需要训练 $c(c-1)/2$ 个 SVM。
- 一对多法：对第 i 个类，将其余 $c-1$ 个类作为负类，训练一个 SVM，共训练 c 个 SVM。
- DAG 法：构造一个不断排除可能性的 DAG，对其每个分支节点训练一个 SVM

SVM 的标准型是一个不等式约束的二次优化问题，可以利用第2.2节介绍的 Lagrange 对偶转化成对偶问题。由于满足 Slater 条件，对偶问题和原问题等价，对偶问题最优解即原问题最优解。写出对偶问题为

$$\begin{aligned} \max_{\alpha} \min_{w,b} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N} \alpha_i (1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)) \\ \text{s.t.} \quad & \alpha_i \geq 0, i = 1, 2, ..., N \end{aligned} \tag{3.14}$$

求解内层最小化问题，该问题是一个二次无约束优化问题，直接对变量求偏导并令其为 0，可得最优解的约束条件

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{m} \alpha_i \left(1 - y^{(i)} \left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right)\right) \right) \\ &= \mathbf{w} + \sum_{i=1}^{m} \alpha_i \left(-y^{(i)}\mathbf{x}^{(i)}\right) \\ &= \mathbf{w} - \sum_{i=1}^{m} \alpha_i y^{(i)}\mathbf{x}^{(i)} = \mathbf{0} \\ \frac{\partial \mathcal{L}}{\partial b} &= \frac{\partial}{\partial b} \left(\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{m} \alpha_i \left(1 - y^{(i)} \left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right)\right) \right) \\ &= \sum_{i=1}^{m} \alpha_i \left(-y^{(i)}\right) \\ &= -\sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \end{aligned}$$

利用这两个约束条件，可以化简最优函数值

$$
\begin{aligned}
\mathcal{L}\left(\mathbf{w}^{\star}, b^{\star}, \alpha\right) &= \frac{1}{2}\mathbf{w}^{\star T}\mathbf{w}^{\star} + \sum_{i=1}^{m}\alpha_i^{\star}(1 - y^{(i)}(\mathbf{w}^{\star T}\mathbf{x}^{(i)} + b)) \\
&= \frac{1}{2}\mathbf{w}^{\star T}\mathbf{w}^{\star} + \sum_{i=1}^{m}\alpha_i - \mathbf{w}^{\star T}\sum_{i=1}^{m}\alpha_i y^{(i)}\mathbf{x}^{(i)} - b\sum_{i=1}^{m}\alpha_i y^{(i)} \\
&= \frac{1}{2}\mathbf{w}^{\star T}\mathbf{w}^{\star} + \sum_{i=1}^{m}\alpha_i - \mathbf{w}^{\star T}\mathbf{w}^{\star} - b\cdot 0 \\
&= -\frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y^{(i)}\mathbf{x}^{(i)}\right)^{T}\left(\sum_{j=1}^{m}\alpha_j y^{(j)}\mathbf{x}^{(j)}\right) + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}{\mathbf{x}^{(i)}}^{T}\mathbf{x}^{(j)} + \sum_{i=1}^{m}\alpha_i
\end{aligned}
$$

写成最小化的形式，于是得到 **SVM 对偶型**

$$
\begin{aligned}
\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}{\mathbf{x}^{(i)}}^{T}\mathbf{x}^{(j)} - \sum_{i=1}^{m}\alpha_i \\
\text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, 2, \ldots, m \\
& \sum_{i=1}^{m}\alpha_i y^{(i)} = 0
\end{aligned}
\tag{3.15}
$$

对偶型最优解 $\alpha^{\star}$ 和基本型最优解 $\mathbf{w}^{\star}$ 关系为

$$
\mathbf{w}^{\star} = \sum_{i=1}^{m}\alpha_i^{\star}y^{(i)}\mathbf{x}^{(i)} \tag{3.16}
$$

这里，系数 w 被表示为了支持向量的线性组合，因此该方法叫做支持向量机。

3.2.2 核 SVM

对于非线性可分二分类问题，我们使用核技巧，将样本嵌入到高维核空间后进行分类，即使用非线性函数 $\phi : R^n \to R^d$ 作用于 $\mathbf{x}$。同时，实际计算中，往往不会直接表示嵌入向量 $\phi(\mathbf{x})$，而是通过核函数 $\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ 计算嵌入向量的内积，而不显示计算嵌入向量，这样可以减少计算量。

算法	**核支持向量机**
算法简述	可分二分类问题中，求解到所有样本最小间隔最大的超曲面
已知	样本 $X \in R^{n\times N}$（不扩充），标签 $Y \in 0, 1^N$
求	$\mathbf{w} \in R^n, b \in R$ 使得 $\min_{w,b}\frac{1}{2}\mathbf{w}^T\mathbf{w}$ 使得 $y^{(i)}(\mathbf{w}^T\phi(\mathbf{x}^{(i)}) + b) \geq 1, i = 1, ..., N$
解类型	迭代解
求解方法	在本节第 4 小节一并描述

常用的核：

- 线性核：$\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = {\mathbf{x}^{(i)}}^{T}\mathbf{x}^{(j)}$
- 多项式核：$\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = (\gamma{\mathbf{x}^{(i)}}^{T}\mathbf{x}^{(j)} + c)^k$

- 高斯核/RBF 核：$\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left\{\frac{||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2}{2\sigma^2}\right\}$

注：若样本集不存在重合样本，则使用高斯核一定可以将数据集分开。该结论证明可以看《SVM 保姆级教程》。

3.2.3 软间隔 SVM

SVM 要求样本在样本空间或嵌入空间中分布在超平面两侧，且离超平面的距离一定要大于最小间隔的一半。但如果数据有噪声，可以适当允许一些样本落在最小间隔之内，但仍然保证分类正确。这样可以增加模型的鲁棒性，防止过拟合。

具体来说，对之前的每个不等式约束，我们引入松弛变量 $\xi_i \geq 0$:

$$y^{(i)}(\mathbf{w}^T\phi(\mathbf{x}^{(i)}) + b) \geq 1 - \xi_i \tag{3.17}$$

同时在优化目标中加入对其的惩罚

$$\min_{w,b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{3.18}$$

这样，当样本仍在间隔之外时，惩罚项不会生效。样本处于间隔内时，越靠近超平面惩罚越大。因此我们有软间隔核 SVM 的标准型

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y^{(i)}(\mathbf{w}^T\phi(\mathbf{x}^{(i)}) + b) \geq 1 - \xi_i, i = 1, ..., N \\ & \xi_i \geq 0, i = 1, ..., N \end{aligned} \tag{3.19}$$

算法	**软间隔核 SVM**
算法简述	可分二分类问题中，求解到所有样本最小间隔最大的超曲面， 但容许一部分样本在最小间隔内
已知	样本 $X \in R^{n\times N}$（不扩充），标签 $Y \in 0, 1^N$
求	$\mathbf{w} \in R^n, b \in R$ 使得 $\min_{w,b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i$ 使得 $y^{(i)}(\mathbf{w}^T\phi(\mathbf{x}^{(i)}) + b) \geq 1 - \xi_i, i = 1, ..., N$ 且 $\xi_i \geq 0, i = 1, ..., N$
解类型	迭代解
求解方法	在本节第 4 小节一并描述

类似地，利用 Lagrange 对偶方法，可以写出软间隔核 SVM 对应的对偶形式为

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^{m}\alpha_i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, m \\ & \sum_{i=1}^{m}\alpha_i y_i = 0 \end{aligned} \tag{3.20}$$

详细的推导过程可见《SVM 保姆级教程》。

3.2.4 求解方法

由于线性 SVM 和核 SVM 都是 4.3 节中软间隔核 SVM 的特例，因此下面只讨论软间隔核 SVM 的优化求解方法。

Pegasos 算法

Pegasos 算法是一种梯度下降算法，将基本型中的不等式约束通过合页损失合并到损失函数中，然后对参数梯度进行求解，进行一阶更新。

DCD 算法

DCD 算法全称为 Dual Coordinate Descent，即对偶坐标下降。这是一种利用坐标下降法的优化算法。所谓坐标下降，就是在优化迭代的过程中一次只变化一个维度，而不是梯度下降的全部维度。

DCD 算法基本思路如下。首先，将样本维度扩充，将偏置项 b 合并到 $\mathbf{w}$ 中。这样，对偶型中就没有了 $\sum_{i=1}^{m}\alpha_i y_i = 0$ 的约束。此时，α 向量的每一个分量就可以分别优化。对于优化的每一步，仅关注分量 α_i。可以将目标函数中和 α_i 相关的所有项提出来，其他无关项可省略，是一个关于 α_i 的二次函数。约束中只有一个区间约束。在这个区间上求最优值，然后更新即可。

SMO 算法

SMO 算法全称为 Sequential Minimal Optimization，即序列最小优化。和 DCD 算法类似，SMO 算法也是选取参数分量分别优化。不过不同的是，SMO 一次优化两个变量，此时仍然是凸二次优化问题。SMO 的具体实现细节待更新。

3.3 其他回归方法

3.3.1 最小二乘回归

第 3 章中介绍了线性回归的最小二乘方法，它相当于求目标函数在线性基函数 $x_1, x_2, ..., x_n$ 上的投影。事实上对于非线性函数，可以使用非线性基来进行拟合，如傅里叶基、勒让德多项式基、切比雪夫多项式基、径向基等等。

假设有基函数 $\phi(x) = [\phi_1(x), \phi_2(x), ..., \phi_n(x)]^T$，满足 $< \phi_i(x), \phi_j(x) >= 0, i \neq j$。假设数据集中的样本排成矩阵 $\phi(X) = [\phi(x_1), \phi(x_2), ..., \phi(x_n)]$。

算法	**非线性最小二乘回归**
算法简述	对于回归问题，假设样本和标签服从非线性关系，样本可以展开为一组基，求出使平方误差最优的回归系数
已知	样本集 $\phi(X) \in R^{n \times N}$，标签 $\mathbf{y} \in R^N$
求	线性系数 $\mathbf{w} \in R^n$ 使得 $\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2}\|\|\phi(X)^T\mathbf{w} - \mathbf{y}\|\|^2$
解类型	闭式解
闭式解	$\mathbf{w}^\star = (\phi(X)^T\phi(X))^{-1}\phi(X)^T\mathbf{y}$

3.3.2 k 近邻回归

k 近邻回归是一种局部拟合算法，它简单的将离自己最近的 k 个邻居的函数值取平均作为预测值。这种算法拟合出的函数不光滑，且数据量越少的地方越不光滑。

算法	**k 近邻回归**
算法简述	对于回归问题，使用 k 近邻的标签的均值作为回归值
已知	样本 $X \in R^{n\times N}$，标签 $\mathbf{y} \in R^N$
求	$f : R^n \to R$ 使得 $f(\mathbf{x})$ 尽量好的拟合 $\mathbf{y}$
解类型	确定解
求解算法	$f(\mathbf{x}) = \frac{\sum_{i=1}^{K} y^{(k_i)}}{K}$, 其中 $\mathbf{x}^{(k_i)}$ 是距离 $\mathbf{x}$ 最近的前 K 个样本

值得注意的是，k 近邻方法经常被用于构建数据的邻接图，即将每个样本作为一个节点，样本和 k 近邻之间用边连接的图。邻接图不依赖标签，在降维和聚类问题中能够发挥巨大作用。

3.3.3 核回归

核回归是对 k 近邻回归进行平滑化的一种算法，由于核函数是连续的，拟合出的曲线也较为连续。距离加权回归可以看作核回归的一个特例。

算法	**核回归**
算法简述	对于回归问题，使用核函数计算权值对已有数据值进行加权
已知	样本 $X \in R^{n\times N}$，标签 $\mathbf{y} \in R^N$
求	$f : R^n \to R$ 使得 $f(\mathbf{x})$ 尽量好的拟合 y
解类型	闭式解
闭式解	$f(\mathbf{x}) = \frac{\sum_{i=1}^{N} \kappa(\mathbf{x}, \mathbf{x}^{(i)}) y^{(i)}}{\sum_{i=1}^{N} \kappa(\mathbf{x}, \mathbf{x}^{(i)})}$

3.3.4 支持向量回归 SVR

支持向量机不仅可以用于分类，也可以用于回归。不过不同的是，用于回归时应当使距离超平面最远的样本的间隔最大，使间隔包络住样本点；同时限制最大间隔，防止回归方向偏离。支持向量回归的求解和 SVM 一样，也是化成对偶形式再对参数依次优化。

支持向量回归只能处理线性回归问题。如果数据分布非线性，可以考虑核支持向量回归。

算法	**支持向量回归**
算法简述	线性回归问题中，使距离超平面最远的样本的间隔最大
已知	样本 $X \in R^{n\times N}$，标签 $\mathbf{y} \in R^N$
求	$\mathbf{w} \in R^n, b \in R$ 使得 $\min_{w,b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}(\xi_i^\vee + \xi_i^\wedge)$ 使得 $-\epsilon - \xi_i^\vee \le y^{(i)} - (\mathbf{w}^T\phi(\mathbf{x}^{(i)}) + b) \le \epsilon + \xi_i^\wedge, i = 1, ..., N$ 且 $\xi_i^\vee \ge 0, \xi_i^\wedge \ge 0, i = 1, ..., N$
解类型	迭代解
求解方法	化为对偶问题，SMO 等方法求解

3.4 其他分类方法

3.4.1 Fisher 线性判别

算法	**Fisher 线性判别**
算法简述	对于二分类问题，寻找最优投影直线， 使样本点在该直线上类内间距最小，类间间距最大
已知	样本 $X \in R^{n\times N}$，标签 $\mathbf{y}^{(i)}$
求	投影方向 $w^\star \in R^n$，使得 $\max_{\mathbf{w}} J_F(\mathbf{w})$
解类型	闭式解
闭式解	$\mathbf{w}^\star = S_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$

对于两分类问题，我们可以对数据集求出各类的均值

$$\mathbf{m}_i = \frac{\sum_{\mathbf{x}\in\Omega_i} \mathbf{x}}{|\Omega_i|} \tag{3.21}$$

以及类内离散度矩阵

$$S_i = \sum_{\mathbf{x}\in\Omega_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \tag{3.22}$$

可得总类内离散度矩阵

$$S_w = S_1 + S_2 \tag{3.23}$$

同时有类间离散度矩阵

$$S_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \tag{3.24}$$

设向量 $\mathbf{w}$ 满足 $\mathbf{w}^T\mathbf{w} = 1$，则向量 $\mathbf{x}$ 在该向量上投影长度为 $\mathbf{w}^T\mathbf{x}$。据此，可以计算投影直线上的类内、类间离散度矩阵

$$\begin{aligned} \bar{\mathbf{m}}_i &= \frac{\sum_{\mathbf{x}\in\Omega_i} \mathbf{w}^T\mathbf{x}}{|\Omega_i|} \\ \bar{S}_i^2 &= \sum_{\mathbf{x}\in\Omega_i} (\mathbf{w}^T\mathbf{x} - \bar{\mathbf{m}}_i)^2 \\ \bar{S}_w &= \bar{S}_1^2 + \bar{S}_2^2 \\ \bar{S}_b &= (\bar{\mathbf{m}}_1 - \bar{\mathbf{m}}_2)^2 \end{aligned} \tag{3.25}$$

定义 Fisher 准则函数

$$J_F(\mathbf{w}) = \frac{\bar{S}_b}{\bar{S}_w} = \frac{(\bar{\mathbf{m}}_1 - \bar{\mathbf{m}}_2)^2}{\bar{S}_1^2 + \bar{S}_2^2} \tag{3.26}$$

闭式解推导：

$$
\begin{aligned}
J_F(\mathbf{w}) =& \frac{(\bar{\mathbf{m}}_1 - \bar{\mathbf{m}}_2)^2}{\bar{S}_1^2 + \bar{S}_2^2} \\
=& \frac{\mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w}}{\mathbf{w}^T\left[\sum_{i=1,2}\sum_{\mathbf{x}\in\Omega_i}(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T\right]\mathbf{w}} \\
=& \frac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}
\end{aligned}
$$

注意此处 w 无需满足模长为 1 约束，因为上下模长项会相消。为保证分母非零，设分母为一常数，即引入约束 $w^T S_w w = c \neq 0$。有

$$
\begin{aligned}
\min_{\mathbf{w}} \quad & \mathbf{w}^T S_b \mathbf{w} \\
\text{s.t.} \quad & w^T S_w w = c
\end{aligned}
\tag{3.27}
$$

定义 Lagrange 函数

$$
\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^T S_b \mathbf{w} + \lambda(\mathbf{w}^T S_w \mathbf{w} - c)
$$

对 $\mathbf{w}$ 求偏导令其为 0，有

$$
S_b \mathbf{w}^\star = \lambda S_w \mathbf{w}^\star
$$

$$
S_w^{-1} S_b \mathbf{w}^\star = \lambda \mathbf{w}^\star
$$

由

$$
S_b \mathbf{w}^\star = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{w}^\star = R(\mathbf{m}_1 - \mathbf{m}_2)
$$

可知 $S_b\mathbf{w}^\star$ 和 $\mathbf{m}_1 - \mathbf{m}_2$ 同向。由于不关心 $\mathbf{w}$ 的长度，可去除所有标量项，取

$$
\mathbf{w}^\star = S_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \tag{3.28}
$$

3.4.2 k 近邻分类

除了回归问题，k 近邻方法也可以用于分类问题。这是一种典型的非参数模型，事实上是依靠样本将参数分布在整个样本空间中。

算法	**k 近邻分类**
算法简述	对于分类问题，使用 k 近邻的标签的众数作为类别
已知	样本 $X \in R^{n\times N}$，标签 $\mathbf{y} \in R^N$
求	$f : R^n \to L_c = 1, 2, ..., c$ 使得 $f(\mathbf{x})$ 尽量接近真实类别标签
解类型	确定解
求解算法	$f(\mathbf{x}) = mode_{t\in kNN(\mathbf{x})}(t)$，其中 $kNN(\mathbf{x})$ 是距离 $\mathbf{x}$ 最近的前 K 个样本

4 降维方法

降维是一类应用广泛的问题。它的本质是压缩编码或特征提取。所谓压缩编码，就是构建一个编码器，将样本编码到隐空间向量；再构造一个解码器，将隐空间向量恢复为高维数据。可以在有监督或无监督的情况下进行降维。本章先介绍通过线性变换进行降维的方法，再介绍非线性的降维方法，最后介绍一类借助神经网络设计的降维方法——自编码器。

4.1 线性降维方法

4.1.1 主成分分析 PCA

PCA 是最常用的线性降维算法。它本质上是在空间中寻找 d 个坐标轴，使样本在其上的投影方差依次最大。

算法	**主成分分析**
算法简述	对于降维问题，依次选取 d 个样本方差最大的 正交投影方向作为基，得到降维后向量
已知	样本 $X \in R^{n\times N}$
求	线性变换 $\hat{P} \in R^{n\times d}$ 使得 $X' = X\hat{P}$ 方差最大
解类型	闭式解
算法步骤	1. 将每个样本减去样本均值 $\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} - \bar{\mathbf{x}}$ 2. 计算维度协方差矩阵 $C = \tilde{X}\tilde{X}^T \in R^{n\times n}$ 3. 协方差矩阵进行相似对角化分解 $C = Q^T\Lambda Q$ 4. 使用单位矩阵 R 将 Λ 重排序为 $\tilde{\Lambda}$， 使得 $\tilde{\Lambda}$ 的特征值依序从大到小. 即 $\Lambda = R^T\tilde{\Lambda}R$ 5. 取 R^TQ^T 的前 d 列为 $\hat{P}$

4.1.2 局部保持投影 LPP

算法	**局部保持投影**
算法简述	对于线性降维问题，最小化以原空间距离加权的投影后样本距离
已知	样本 $X \in R^{n\times N}$
求	线性变换 $A \in R^{d\times n}$ 使得加权投影后样本距离 $J(A) = \mathrm{tr}(A^TX^tLXA)$ 最小
解类型	闭式解
算法步骤	1. 使用 k 近邻算法构建邻接图（无方向） 2. 对相邻边计算权重 $w_{ij} = \exp\{-\frac{\|\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\|^2}{\sigma^2}\}$ 3. 计算度矩阵 D 和拉普拉斯矩阵 L 4. 取 $(X^TDX)^{-1}X^TLX$ 的最小 d 个特征值对应特征向量为 A

LPP 也是一种线性降维方法，它的算法目标也是找一个矩阵 $A \in R^{d\times n}$ 将 X 投影到 $Y = AX$。和 PCA 不同，LPP 的优化目标是带权重的投影后样本距离

$$J(A) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}||\mathbf{y}^{(i)} - \mathbf{y}^{(j)}||^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}||A\mathbf{x}^{(i)} - A\mathbf{x}^{(j)}||^2 \tag{4.1}$$

其中 w_{ij} 是和高维空间距离相关的权重，距离越近权重越大，距离越远权重越小。具体来说，算法首先寻找每个样本 $\mathbf{x}^{(i)}$ 的 k 近邻，将近邻之间的权重置为

$$w_{ij} = \exp\left\{-\frac{||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2}{\sigma^2}\right\}$$

即径向基函数。其余样本间权重一律为 0. 事实上，此时各样本已经组成了一个带权图。除了权重矩阵 W，还可以计算度矩阵 D 和拉普拉斯矩阵 L。

对优化目标进行展开

$$\begin{aligned}
J(A) =& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}||A\mathbf{x}^{(i)} - A\mathbf{x}^{(j)}||^2 \\
=& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(A\mathbf{x}^{(i)} - A\mathbf{x}^{(j)})^T(A\mathbf{x}^{(i)} - A\mathbf{x}^{(j)}) \\
=& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}({\mathbf{x}^{(i)}}^T A^T A\mathbf{x}^{(i)} + {\mathbf{x}^{(j)}}^T A^T A\mathbf{x}^{(j)} - 2{\mathbf{x}^{(i)}}^T A^T A\mathbf{x}^{(j)}) \\
=& \sum_{i=1}^{N} d_{ii}{\mathbf{x}^{(i)}}^T A^T A\mathbf{x}^{(i)} - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}{\mathbf{x}^{(i)}}^T A^T A\mathbf{x}^{(j)} \\
=& \operatorname{tr}(A^T X^T (D - W) X A) \\
=& \operatorname{tr}(A^T X^T L X A)
\end{aligned}$$

由于投影矩阵 A 有无尺度性，因此增加一个关于投影后向量模长的约束

$$A^T X^T D X A = I$$

即

$$\begin{aligned}
\min_{A} \quad & J(A) = \operatorname{tr}(A^T X^T L X A) \\
\text{s.t.} \quad & A^T X^T D X A = I
\end{aligned} \tag{4.2}$$

使用 Lagrange 乘子法，得到 Lagrange 函数

$$\mathcal{L}(A, \Lambda) = \operatorname{tr}(A^T X^T L X A) + \operatorname{tr}(\Lambda(A^T X^T D X A - I))$$

求偏导

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial A} = 2X^T L X A - 2X^T D X A \Lambda = 0 \\
&X^T L X A = X^T D X A \Lambda \\
&(X^T D X)^{-1} X^T L X A = A\Lambda
\end{aligned}$$

即 A 的列向量是矩阵 $(X^T DX)^{-1}X^T LX$ 的特征向量。为最小化目标函数，取最小的 d 个特征值对应的特征向量组成 A 即可。

注：LPP 和后面列出的 LE 方法一脉相承，LPP 本质是在 LE 的基础上将变换线性化，可能更加不精确，但可以学到变换而不只是嵌入。

4.2 非线性降维方法

4.2.1 核 PCA

低维空间中的非线性复杂分布往往可以在映射到高维空间后变得简单。核 PCA 的思想就是在核函数衍生出的高维空间中对 $\phi(\mathbf{x})$ 进行降维，从而获得非线性空间中的主成分。然而，一般的核函数不能显式提供 $\phi(\mathbf{x})$，强行计算的成本也高。因此需要使用变换技巧将其都转化成核函数的计算。

算法	**核主成分分析**
算法简述	对于降维问题，在核空间中选取 d 个样本方差最大的正交投影方向 作为基，得到降维后向量
已知	样本 $X \in R^{n\times N}$
求	变换 $f: R^n \to R^d$ 使得 $X' = f(X)$ 方差最大
解类型	闭式解
算法步骤	1. 计算样本核矩阵 K: $k_{ij} = \kappa(x^{(i)}, x^{(j)})$ 2. 计算中心化核矩阵 $\tilde{K} = K - K\mathbf{1}_N - \mathbf{1}_N K + \mathbf{1}_N K \mathbf{1}_N$ 3. 对中心化核阵进行特征分解 $\tilde{K}P = \Lambda P$, $\Lambda = \mathrm{diag}\{\lambda_1, \lambda_2, ..., \lambda_n\}$ 是从大到小排序后的特征值矩阵，$P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ 4. 计算降维后样本 $X' = P_d^T \tilde{K}$，其中 $P_d = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_d\}$ 为 P 的前 d 列

首先要对 $\phi(\mathbf{x})$ 进行去中心化，使其均值为 0. 虽然我们无法直接计算嵌入向量，但我们只关心嵌入向量的 Gram 矩阵，因此我们需要的是去中心化向量 $\tilde{\mathbf{x}}$ 的核矩阵 $\tilde{K}$。

设 $\phi(X) = [\phi(\mathbf{x}^{(1)}), \phi(\mathbf{x}^{(1)}), ..., \phi(\mathbf{x}^{(1)})] \in R^{m\times N} (m > n)$. 设元素全为 $\frac{1}{N}$ 的 $N \times N$ 矩阵为 $\mathbf{1}_N$，设元素全为 1 的 n 维列向量为 $\mathbf{e}_N$，则有核矩阵

$$K = \phi(X)^T \phi(X)$$

嵌入向量均值

$$\bar{\phi}(\mathbf{x}) = \frac{\sum_{i=1}^N \phi(\mathbf{x}^{(i)})}{N} = \frac{1}{N}\phi(X)\mathbf{e}_N$$

去中心化向量

$$\tilde{\phi}(X) = \phi(X) - \bar{\phi}(\mathbf{x})\mathbf{e}_N = (I - \frac{1}{N}\mathbf{e}_N\mathbf{e}_N^T)\phi(X) = (I - \mathbf{1}_N)\phi(X)$$

去中心化 Gram 矩阵（去中心化核矩阵）

$$\begin{aligned}
\tilde{K} =& \tilde{\phi}(X)^T \tilde{\phi}(X) \\
=& (I - \mathbf{1}_N)^T \phi(X)^T \phi(X)(I - \mathbf{1}_N) \\
=& (I - \mathbf{1}_N)^T K (I - \mathbf{1}_N) \\
=& K - \mathbf{1}_N K - K\mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N
\end{aligned}$$

我们需要在核空间中找到 d 个正交的单位向量 $\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_d$ 作为基，于是 $\phi(\mathbf{x})$ 在这组基上的投影就构成了降维后的 R^d 坐标向量。

$$\begin{aligned}
&\max_{\mathbf{p}} \sum_{i=1}^{N} \left\langle \mathbf{p}, \tilde{\phi}(\mathbf{x}^{(i)}) \right\rangle^2 \\
=&\max_{\mathbf{p}} \sum_{i=1}^{N} (\mathbf{p}^T \tilde{\phi}(\mathbf{x}^{(i)}))^2 \\
=&\max_{\mathbf{p}} \mathbf{p}^T \left(\sum_{i=1}^{N} \tilde{\phi}(\mathbf{x}^{(i)}) \tilde{\phi}(\mathbf{x}^{(i)}) \right) \mathbf{p} \\
=&\max_{\mathbf{p}} \mathbf{p}^T C_e \mathbf{p}
\end{aligned}$$

其中中间的矩阵记作 C_e，它是嵌入后的协方差矩阵。

$$C_e = \sum_{i=1}^{N} \tilde{\phi}(\mathbf{x}^{(i)}) \tilde{\phi}(\mathbf{x}^{(i)}) = \tilde{\phi}(X) \tilde{\phi}(X)^T$$

正如 PCA 算法中所推导的，当要求一组相互正交的 p 使每个 p 都让二次型目标取极值时，应当取 C_e 的对应最大的几个特征值的特征向量，即

$$C_e \mathbf{p} = \tilde{\phi}(X) \tilde{\phi}(X)^T \mathbf{p} = \lambda \mathbf{p}$$

然而，协方差矩阵 C_e 的值可能是无法计算或计算成本很大的，因此无法直接相似对角化。将上式变形

$$\mathbf{p} = \frac{1}{\lambda} \tilde{\phi}(X) \tilde{\phi}(X)^T \mathbf{p} = \frac{1}{\lambda} \tilde{\phi}(X) \alpha \tag{4.3}$$

其中

$$\alpha = \tilde{\phi}(X)^T \mathbf{p} \tag{4.4}$$

是一个 N 维列向量。也就是说，根据式4.3，$\mathbf{p}$ 是各 $\phi(\mathbf{x})$ 的线性组合。代入式4.3中

$$\frac{1}{\lambda} \tilde{\phi}(X) \tilde{\phi}(X)^T \tilde{\phi}(X) \alpha = \tilde{\phi}(X) \alpha$$

两边左乘 $\tilde{\phi}(X)^T$，有

$$\frac{1}{\lambda} \tilde{\phi}(X)^T \tilde{\phi}(X) \tilde{\phi}(X)^T \tilde{\phi}(X) \alpha = \tilde{\phi}(X)^T \tilde{\phi}(X) \alpha$$

即

$$\frac{1}{\lambda} \tilde{K} \tilde{K} \alpha = \tilde{K} \alpha$$

$$\tilde{K} \alpha = \lambda \alpha$$

因此，$\tilde{K}\alpha$ 是 $\tilde{K}$ 的特征向量，即 α 是 $\tilde{K}$ 的特征向量。这样，我们可以由 $\tilde{K}$ 的前 d 大的特征值得到 d 个 $\mathbf{p}$ 对应的 α。

接下来只需要将 $\tilde{\phi}(X)$ 投影到 $P = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_d]$ 上即可。为了投影方便，需要将 P 的列向量归一化，即

$$\mathbf{p}^T\mathbf{p} = 1$$

由式4.3，有

$$(\frac{1}{\lambda}\tilde{\phi}(X)\alpha)^T(\frac{1}{\lambda}\tilde{\phi}(X)\alpha) = 1$$

$$\frac{1}{\lambda^2}\alpha^T\tilde{\phi}(X)^T\tilde{\phi}(X)\alpha = 1$$

$$\frac{1}{\lambda^2}\alpha^T\tilde{K}\alpha = 1$$

$$\frac{1}{\lambda^2}\alpha^T\lambda\alpha = 1$$

$$\alpha^T\alpha = \lambda$$

因此只要放缩 α 使其模长为 $\sqrt{\lambda}$, 就可以使 $\mathbf{p}$ 归一化。此时有

$$f(\mathbf{X}) = (\tilde{\phi}(X)^T P)^T = [\alpha_1, \alpha_2, ..., \alpha_d]^T$$

即，样本 $\mathbf{x}^{(i)}$ 的降维结果就是求出的各 α 的第 i 行元素组成的 d 维向量。

4.2.2 多维度缩放 MDS

算法	**多维度缩放**
算法简述	对于降维问题，保持降维后各样本间距不变
已知	样本 $X \in R^{n\times N}$
求	$f(X)$，其中变换 $f: R^n \to R^d$ 使得 $\|\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\| = \|\|f(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(j)})\|\|$
解类型	闭式解
算法步骤	1. 计算原空间中样本两两间距 $d_{ij} = \|\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\|$ 2. 计算矩阵 B：$b_{ij} = -\frac{1}{2}(d_{ij}^2 - \frac{1}{N}\sum_{k=1}^N d_{kj}^2 - \frac{1}{N}\sum_{l=1}^N d_{il}^2 + \frac{1}{N^2}\sum_{k=1}^N\sum_{l=1}^N d_{kl}^2)$ 3. 对 B 矩阵做相似对角化分解 $B = P^T\Lambda P$， 取 P^T 中对应前 d 大特征值的 d 行组成 P_d， 对应特征值组成 Λ_d，$f(X) = P_d\Lambda_d^{\frac{1}{2}} \in R^{d\times N}$ 即为所求

假设样本 $X \in R^{n\times N}$ 降维后变成 $f(X) = Z = [\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(N)}] \in R^{d\times N}$。可以写出降维后内积矩阵

$$B = Z^T Z$$

其中 $b_{ij} = {\mathbf{z}^{(i)}}^T\mathbf{z}^{(j)}$。设原空间中样本两两间距

$$d_{ij} = ||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||$$

有距离约束

$$
\begin{aligned}
d_{ij}^2 =& ||\mathbf{z}^{(i)} - \mathbf{z}^{(j)}||^2 \\
=& {\mathbf{z}^{(i)}}^T \mathbf{z}^{(i)} - 2{\mathbf{z}^{(i)}}^T \mathbf{z}^{(j)} + {\mathbf{z}^{(j)}}^T \mathbf{z}^{(j)} \\
=& b_{ii} - 2b_{ij} + b_{jj}
\end{aligned} \tag{4.5}
$$

不妨设降维后样本是中心化的，即

$$
\sum_{i=1}^{N} \mathbf{z}^{(i)} = \mathbf{0}
$$

则 B 矩阵有行和列和为 0 的约束

$$
\begin{aligned}
\sum_{i=0}^{N} b_{ij} =& 0 \\
\sum_{j=0}^{N} b_{ij} =& 0
\end{aligned}
$$

此时考虑 D 矩阵的行列平方和

$$
\begin{aligned}
d_{\cdot\, j}^2 =& \sum_{i=0}^{N} d_{ij}^2 = \operatorname{tr}(B) + N b_{jj} \\
d_{i\,\cdot}^2 =& \sum_{j=0}^{N} d_{ij}^2 = \operatorname{tr}(B) + N b_{ii} \\
d_{\cdot\,\cdot}^2 =& \sum_{i=0}^{N} \sum_{j=0}^{N} d_{ij}^2 = 2N \operatorname{tr}(B)
\end{aligned}
$$

代入式4.5中，即可求出 B

$$
\begin{aligned}
b_{ij} =& -\frac{1}{2}(d_{ij}^2 - b_{ii} - b_{jj}) \\
=& -\frac{1}{2}(d_{ij}^2 - \frac{1}{N}(d_{i\,\cdot}^2 - \frac{1}{2N} d_{\cdot\,\cdot}^2) - \frac{1}{N}(d_{\cdot\, j}^2 - \frac{1}{2N} d_{\cdot\,\cdot}^2)) \\
=& -\frac{1}{2}(d_{ij}^2 - \frac{1}{N} d_{i\,\cdot}^2 - \frac{1}{N} d_{\cdot\, j}^2 + \frac{1}{N^2} d_{\cdot\,\cdot}^2)
\end{aligned} \tag{4.6}
$$

对 B 矩阵做相似对角化分解 $B = P^T \Lambda P$，取 P 中对应前 k 大特征值的 d 列组成 P_d，对应特征值组成 Λ_d，即可得到 $Z = \Lambda_d^{\frac{1}{2}} P_d$

4.2.3 等距特征映射 ISOMAP

MDS 方法保留了样本在高维空间中的距离，但有时数据是分布在高维空间中的低维流形上的，此时应当尝试保留流形上的距离。ISOMAP 通过流形最短路径估计这个距离。

算法	**等距特征映射**
算法简述	对于降维问题，保持降维后各样本间距不变
已知	样本 $X \in R^{n\times N}$
求	变换 $f: R^n \to R^d$ 使得 $\|\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\| = \|\|f(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(j)})\|\|$
解类型	确定解
算法步骤	1. 计算高维空间中样本的 k 近邻，将其对称化，保证形成连通图 2. 将每个样本和其 k 近邻的流形距离设置为其欧氏距离 3. 对于非近邻点对，使用 Dijkstra 算法求出最短距离作为流形距离 4. 以样本间流形距离作为 D 矩阵，应用 MDS 算法，得到降维结果

由于几乎每对点之间都要计算最短距离，且 Dijkstra 最短距离的计算复杂度是 $O(N\log N)$（邻接图边数和节点数成线性关系），所以 ISOMAP 算法总时间复杂度至少为 $O(N^3\log N)$。对于数据量较大的情况，最好避免使用这个算法。

4.2.4 局部线性嵌入 LLE

算法	**局部线性嵌入**
算法简述	对于降维问题，保持降维后各样本间距不变
已知	样本 $X \in R^{n\times N}$
求	回归系数 $W \in R^{N\times N}$ 使得 $\min J_i(W) = \|\|\mathbf{x}^{(i)} - \sum_{j=1}^{N} w_{ij}\mathbf{x}^{(j)}\|\|^2$ 且满足近邻约束 嵌入向量 $Y \in R^{d\times N}$ 使得 $\min J(Y) = \sum_{i=1}^{N} \|\|\mathbf{y}^{(i)} - \sum_{j\in N(i)} w_{ij}\mathbf{y}^{(j)}\|\|^2$
解类型	闭式解
算法步骤	1. 计算高维空间中样本的 k 近邻，将其对称化，保证形成连通图 2. 逐样本计算近邻方差 $C_i = (\mathbf{x}^{(j)} - \mathbf{x}^{(i)})^T(\mathbf{x}^{(k)} - \mathbf{x}^{(i)})$ 3. 计算回归系数 $\tilde{\mathbf{w}}_i = \dfrac{C_i^{-1}\mathbf{1}_N}{\mathbf{1}_N^T C_i^{-1}\mathbf{1}_N}$ 4. 取 $(W-I)(W-I)^T$ 中最小的 d 个特征值对应的特征行向量组成 Y

相比于以上几种方法关注全局性质的保持，LLE 和 LPP 类似，都更关注数据局部性质的保持。LLE 认为，任何一个样本都可以通过近邻样本的线性组合来近似。如果在低维空间中也能保持和高维空间中一样的线性组合关系，就是保留了局部特征。因此 LLE 计算分为两步，首先寻找最优线性组合系数，其次计算最优低维表示。

假设有样本 $X \in R^{n\times N}$。用 kNN 算法可以构建其邻接图（非对称）。设有系数矩阵 $W = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_N] \in R^{N\times N}$。若 $\mathbf{x}^{(j)}$ 不是 $\mathbf{x}^{(i)}$ 的 k 近邻，则 $w_{ij} = 0$。其余系数用于拟合，即

$$\mathbf{x}^{(i)} \approx \sum_{j=1}^{N} w_{ji}\mathbf{x}^{(j)} = X\mathbf{w}_i$$

有约束

$$\sum_{j=1}^{N} w_{ji} = 1$$

最优参数就是最小化回归的平方误差

$$\min_W J(W) = \sum_{i=1}^{N} ||\mathbf{x}^{(i)} - \sum_{j \in N(i)} w_{ji}\mathbf{x}^{(j)}||^2$$

其中 $N(i)$ 表示第 i 个样本的 K 近邻的下标集合。W 的一行只和单个样本有关，记 $\tilde{\mathbf{w}}_i \in R^k$ 为第 i 个样本的 K 近邻的加权系数向量. 考虑第 i 个样本的回归平方误差

$$\begin{aligned}
J_i(W) =& ||\mathbf{x}^{(i)} - \sum_{j=1}^{N} w_{ji}\mathbf{x}^{(j)}||^2 \\
=& ||\sum_{j=1}^{N} w_{ji}(\mathbf{x}^{(i)} - \mathbf{x}^{(j)})||^2 \\
=& \sum_{j=1}^{K}\sum_{k=1}^{K} w_{ji}w_{ki}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})^T(\mathbf{x}^{(k)} - \mathbf{x}^{(i)}) \\
=& \sum_{j=1}^{K}\sum_{k=1}^{K} w_{ji}w_{ki}c_{jk} \\
=& \mathbf{w}_i^T C_i \mathbf{w}_i
\end{aligned}$$

设 $\mathbf{1}_K$ 代表全 1 的 K 维列向量, 约束项可以写为

$$\mathbf{1}_N^T \mathbf{w}_i = 1$$

即

$$\begin{aligned}
\min_{\mathbf{w}_i} \quad & J_i(\mathbf{w}_i) = \mathbf{w}_i^T C_i \mathbf{w}_i \\
\text{s.t.} \quad & \mathbf{1}_N^T \mathbf{w}_i = 1
\end{aligned} \tag{4.7}$$

使用 Lagrange 乘子法

$$\mathcal{L}(\tilde{\mathbf{w}}_i, \lambda) = \tilde{\mathbf{w}}_i^T C_i \tilde{\mathbf{w}}_i + \lambda(\mathbf{1}_N^T \tilde{\mathbf{w}}_i - 1)$$

对 $\tilde{\mathbf{w}}_i$ 求偏导

$$\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{w}}_i} = 2C_i\tilde{\mathbf{w}}_i + \lambda \mathbf{1}_N = 0$$

即

$$\tilde{\mathbf{w}}_i = -\frac{\lambda}{2} C_i^{-1} \mathbf{1}_N$$

使用约束条件归一化，有

$$\tilde{\mathbf{w}}_i = \frac{C_i^{-1}\mathbf{1}_N}{\mathbf{1}_N^T C_i^{-1} \mathbf{1}_N} \tag{4.8}$$

另一方面，设 $\mathbf{x}^{(i)}$ 对应低维表示为 $\mathbf{y}^{(i)}$。优化目标为最小化平方误差

$$\min_Y J(Y) = \sum_{i=1}^{N} ||\mathbf{y}^{(i)} - \sum_{j \in N(i)} w_{ji}\mathbf{y}^{(j)}||^2$$

由于 Y 有非尺度性，加入约束条件

$$\frac{1}{N}YY^T = I$$

重写目标函数为

$$\begin{aligned}
\min J(Y) &= \sum_{i=1}^{N} ||\mathbf{y}^{(i)} - \sum_{j=1}^{N} w_{ji}\mathbf{y}^{(j)}||^2 \\
&= \sum_{i=1}^{N} ||\mathbf{y}^{(i)} - Y\mathbf{w}_i||^2 \\
&= \sum_{i=1}^{N} (\mathbf{y}^{(i)} - Y\mathbf{w}_i)^T(\mathbf{y}^{(i)} - Y\mathbf{w}_i) \\
&= \operatorname{tr}(Y^TY) + \operatorname{tr}(W^TY^TYW) - 2\operatorname{tr}(Y^TYW) \\
&= \operatorname{tr}((W-I)^TY^TY(W-I))
\end{aligned}$$

即

$$\begin{aligned}
\min_{Y} \quad & J(Y) = \operatorname{tr}((W-I)^TY^TY(W-I)) \\
\text{s.t.} \quad & \frac{1}{N}YY^T = I
\end{aligned} \tag{4.9}$$

Lagrange 乘子法

$$\mathcal{L}(Y,\lambda) = \operatorname{tr}((W-I)^TY^TY(W-I)) + \lambda(YY^T - NI))$$

对 Y 求偏导

$$\frac{\partial \mathcal{L}}{\partial Y} = 2Y(W-I)(W-I)^T + 2\lambda Y = 0$$

即

$$Y(W-I)(W-I)^T = \lambda Y$$

取 $(W-I)(W-I)^T$ 中最小的 d 个特征值对应的特征行向量组成 Y 即可。

4.2.5 拉普拉斯特征图 LE

算法	**拉普拉斯特征图**
算法简述	对于降维问题，最小化以原空间距离加权的降维后样本距离
已知	样本 $X \in R^{n\times N}$
求	嵌入向量 $Y \in R^{d\times N}$ 使得使得加权投影后样本距离 $J(Y) = \operatorname{tr}(Y^TLY)$ 最小
解类型	闭式解
算法步骤	1. 使用 k 近邻算法构建邻接图（对称化） 2. 对相邻边计算权重 $w_{ij} = \exp\left\{-\frac{\|\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\|^2}{\sigma^2}\right\}$ 3. 计算度矩阵 D 和拉普拉斯矩阵 L 4. 取 $LD^{-1}L$ 的最小 d 个特征值对应特征行向量为 Y

LE 和 LPP 思路一致，优化目标都是带权重的投影后样本距离，只不过优化变量直接是投影后的向量

$$J(Y)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}||\mathbf{y}^{(i)}-\mathbf{y}^{(j)}||^2=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}||\mathbf{y}^{(i)}-\mathbf{y}^{(j)}||^2$$

其中 w_{ij} 是和高维空间距离相关的权重，距离越近权重越大，距离越远权重越小。具体来说，算法首先寻找每个样本 $\mathbf{x}^{(i)}$ 的 k 近邻，将近邻之间的权重置为

$$w_{ij}=\exp\left\{-\frac{||\mathbf{x}^{(i)}-\mathbf{x}^{(j)}||^2}{\sigma^2}\right\}$$

即径向基函数。其余样本间权重一律为 0. 事实上，此时各样本已经组成了一个带权图。除了权重矩阵 W，还可以计算度矩阵 D 和拉普拉斯矩阵 L。

对优化目标进行展开

$$\begin{aligned}
J(Y)=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}||\mathbf{y}^{(i)}-\mathbf{y}^{(j)}||^2\\
=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}(\mathbf{y}^{(i)}-\mathbf{y}^{(j)})^T(\mathbf{y}^{(i)}-\mathbf{y}^{(j)})\\
=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}({\mathbf{y}^{(i)}}^T\mathbf{y}^{(i)}+{\mathbf{y}^{(j)}}^T\mathbf{y}^{(j)}-2{\mathbf{y}^{(i)}}^T\mathbf{y}^{(j)})\\
=&\sum_{i=1}^{N}d_{ii}{\mathbf{y}^{(i)}}^T\mathbf{y}^{(i)}-\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}{\mathbf{y}^{(i)}}^T\mathbf{Y}^{(j)}\\
=&\operatorname{tr}(Y^TY(D-W))\\
=&\operatorname{tr}(Y^TYL)\\
=&\operatorname{tr}(YLY^T)
\end{aligned}$$

由于 Y 有无尺度性，因此增加一个约束

$$YDY^T=I$$

即

$$\begin{aligned}
\min_{Y}\quad & J(Y)=\operatorname{tr}(YLY^T)\\
\text{s.t.}\quad & YDY^T=I
\end{aligned}\tag{4.10}$$

使用 Lagrange 乘子法，得到 Lagrange 函数

$$\mathcal{L}(Y,\Lambda)=\operatorname{tr}(YLY^T)+\operatorname{tr}(\Lambda(YDY^T-I))$$

求偏导

$$\begin{aligned}
&\frac{\partial\mathcal{L}}{\partial Y}=2YL-2YD\Lambda=0\\
&YL=\Lambda YD\\
&YLD^{-1}=Y\Lambda D
\end{aligned}$$

即 Y 的行向量为 LD^{-1} 矩阵的行特征向量。为最小化目标函数，取最小的 d 个特征值对应的行特征向量组成 Y 即可。

5 聚类方法

5.1 原型聚类

5.1.1 k 均值聚类

k 均值聚类是一种使用 E-M 算法思想（详见第6.2节）的聚类方法。它的算法非常简单，计算量小，但较不稳定，且无法处理非线性环形分布。常用于各种其他聚类算法的初始化。

算法	**k 均值聚类**
算法简述	聚类问题中，迭代更新每个样本点的类别为最近的聚类中心
已知	样本 $X \in R^{n\times N}$（不扩充）
求	分配矩阵 $R \in \{0,1\}^{N\times K}$、聚类均值 $\mu_1, \mu_2, ..., \mu_k \in R^n$， 使得 $\min_R J(R) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}\|\|\mathbf{x}^{(i)} - \mu_j\|\|^2$ ，且 $\sum_{j=1}^{K} r_{ij} = 1, \forall j$
解类型	迭代解
算法步骤	E 步：将每个样本的类别置为最近的聚类中心的类别 M 步：更新每个聚类中心位置为当前类的样本位置均值

5.1.2 模糊 k 均值聚类

k 均值聚类中，分配矩阵是 0-1 矩阵，一个样本能且只能属于一个聚类簇。如果将分配矩阵视为分配权重，使用连续的变量值，就可以使聚类结果更稳定平滑。

算法	**模糊 k 均值聚类**
算法简述	k 均值聚类中，使用连续的分配矩阵
已知	样本 $X \in R^{n\times N}$（不扩充）
求	分配矩阵 $R \in R^{N\times K}, 0 \le r_{ij} \le 1$、聚类均值 $M = [\mu_1, \mu_2, ..., \mu_k] \in R^{N\times K}$， 使得 $\min_{R,M} J(R, M) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}^m\|\|\mathbf{x}^{(i)} - \mu_j\|\|^2$ ，且 $\sum_{j=1}^{K} r_{ij} = 1, \forall j$
解类型	迭代解
算法步骤	E 步：$r_{ij} = \left(\sum_{l=1}^{K}\left(\frac{\|\|\mathbf{x}^{(i)} - \mu_j\|\|^2}{\|\|\mathbf{x}^{(i)} - \mu_k\|\|^2}\right)^{\frac{2}{m-1}}\right)^{-1}$ M 步：$\mu_j = \frac{\sum_{i=1}^{N} r_{ij}^m \mathbf{x}^{(i)}}{\sum_{i=1}^{N} r_{ij}^m}$

设样本 $X \in R^{n\times N}$，分配矩阵 $R \in R^{N\times K}, 0 \le r_{ij} \le 1$，聚类均值 $M = [\mu_1, \mu_2, ..., \mu_k] \in R^{n\times K}$，目标函数

$$\min_{R,M} J(R, M) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}^m\|\|\mathbf{x}^{(i)} - \mu_j\|\|^2$$

对于分配矩阵 R，有约束条件

$$\sum_{j=1}^{K} r_{ij} = 1, \forall j \tag{5.1}$$

即

$$\begin{aligned} \min_{R,M} \quad & J(R,M) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}^{m}||\mathbf{x}^{(i)} - \mu_j||^2 \\ \text{s.t.} \quad & \sum_{j=1}^{K} r_{ij} = 1, \forall j \end{aligned} \tag{5.2}$$

Lagrange 乘子法

$$L(R,M,\lambda) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}^{m}||\mathbf{x}^{(i)} - \mu_j||^2 + \sum_{i=1}^{N}\lambda_i\left(\sum_{j=1}^{K} r_{ij} - 1\right)$$

对分配矩阵求偏导

$$\frac{\partial L}{\partial r_{ij}} = m||\mathbf{x}^{(i)} - \mu_j||^2 r_{ij}^{m-1} + \lambda_j = 0$$

$$r_{ij} = \left(\frac{-\lambda_j}{||m(\mathbf{x}^{(i)} - \mu_j)||^2}\right)^{\frac{1}{m-1}} \tag{5.3}$$

由5.1有

$$\sum_{j=1}^{K}\left(\frac{-\lambda_j}{m||\mathbf{x}^{(i)} - \mu_j||^2}\right)^{\frac{1}{m-1}} = 1$$

即

$$\left(\frac{-\lambda_j}{m}\right)^{\frac{1}{m-1}} = \sum_{j=1}^{K}\left(\frac{1}{||\mathbf{x}^{(i)} - \mu_j||}\right)^{\frac{2}{m-1}}$$

代入式5.3中，有

$$r_{ij} = \frac{1}{\sum_{l=1}^{K}\left(\frac{||\mathbf{x}^{(i)} - \mu_j||}{||\mathbf{x}^{(i)} - \mu_l||}\right)^{\frac{2}{m-1}}} \tag{5.4}$$

对类均值求偏导

$$\frac{\partial L}{\partial \mu_j} = \sum_{i=1}^{N} -2r_{ij}^{m}(\mathbf{x}^{(i)} - \mu_j) = \mathbf{0}$$

$$\mu_j = \sum_{i=1}^{N}\frac{r_i j^m}{\sum_{l=1}^{N} r_{il}^{m}}\mathbf{x}^{(i)} \tag{5.5}$$

迭代求解 R, M 即可

5.1.3 核 k 均值聚类

算法	核 k 均值聚类
算法简述	k 均值聚类中，使用核方法将样本嵌入到高维空间中
已知	样本 $X \in R^{n\times N}$（不扩充）
求	分配矩阵 $R \in \{0,1\}^{N\times K}$ 使得 $\min_R J(R) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}^m ||\phi(\mathbf{x})^{(i)} - \mu_j||^2$ ，且 $\sum_{j=1}^{K} r_{ij} = 1, \forall j$
解类型	迭代解
算法步骤	1. 计算 Gram 矩阵 $K = (\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}))_{ij}$，初始化系数矩阵 $A = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_K] \in N \times K$ 2. (E 步) 计算相似度矩阵 $S = K^T A$ 3. (E 步) 每行中 s_{ij} 最大的位置 $r_{ij} = 1$，其余 $r_{ij} = 0$ 4. (M 步) 更新系数矩阵 $a_{ij} = \frac{r_{ij}}{\sum_{l=1}^{N} r_{il}}$ 5. 若收敛，停止迭代，否则回第 2 步。

前面我们已经多次使用过核技巧，原理不再赘述。核 k-means 方法假设在高维空间中有 K 个聚类中心 $\mu_1, \mu_2, ..., \mu_K$。在更新分配矩阵后，可以计算出每个聚类中心的位置

$$\mu_j = \sum_{i=1}^{N} \frac{r_{ij}\phi(\mathbf{x}^{(i)})}{\sum_{l=1}^{N} r_{il}}$$

由此可以计算每个样本 $\mathbf{x}^{(i)}$ 和高维聚类中心 μ_j 的相似度

$$\begin{aligned} s_{ij} =& \mu_j^T \phi(\mathbf{x}^{(i)}) \\ =& \sum_{k=1}^{N} \frac{r_{kj}\phi(\mathbf{x}^{(k)})^T \phi(\mathbf{x}^{(i)})}{\sum_{l=1}^{N} r_{il}} \\ =& \sum_{k=1}^{N} \frac{r_{kj}\kappa(\mathbf{x}^{(k)}, \mathbf{x}^{(i)})}{\sum_{l=1}^{N} r_{il}} \\ =& \frac{\sum_{k=1}^{N} r_{kj} k_{ki}}{\sum_{l=1}^{N} r_{il}} \end{aligned}$$

设矩阵 $A \in R^{N\times K} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_K]$，且

$$a_{ij} = \frac{r_{ij}}{\sum_{l=1}^{N} r_{il}}$$

则有

$$S = K^T A \tag{5.6}$$

5.2 谱聚类

谱聚类是一种利用降维思想做聚类的方法。它的算法思路是：首先利用 LE 方法通过生成邻接图、保持局部距离来将高维空间变换到低维空间。其次再使用简单聚类方法如 kmeans 进行聚类。

算法	**谱聚类**
算法简述	对于聚类问题，构建邻接图后最小化 NCut 切图权重函数
已知	样本 $X \in R^{n\times N}$
求	分配矩阵 $R \in \{0,1\}^{N\times K}$ 使得切图权重 $NCut(A_1,...,A_k)=\frac{1}{2}\sum_{i=1}^{K}\frac{W(A_i,\bar{A}_i)}{\mathrm{vol}(A_i)}$ 最小
解类型	闭式解
算法步骤	1. 使用 k 近邻算法构建邻接图（对称化） 2. 对相邻边计算权重 $w_{ij}=\exp\left\{-\frac{\|\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|\|^2}{\sigma^2}\right\}$ 3. 计算度矩阵 D 和拉普拉斯矩阵 L 4. 取 $LD^{-1}L$ 的最小 d 个特征值对应特征列向量为 $H \in R^{N\times d}$ 5. 以 H^T 作为降维样本集，进行 k 均值聚类，得到聚类结果 R

不过，谱聚类还有另一种推导思路，从图论的角度出发，也可以得到一样的结果。
首先，我们使用 k 近邻算法将数据集转化为近邻图 $G=\{V,E\}$，近邻边的权重为

$$w_{ij}=\exp\left\{-\frac{||\mathbf{x}^{(i)}-\mathbf{x}^{(j)}||^2}{\sigma^2}\right\} \tag{5.7}$$

计算度矩阵 D 和拉普拉斯矩阵 L，不再赘述。

$$d_{ij}=\begin{cases}0 & j\neq i\\ \sum_{k=1}^{N}w_{ik} & j=i\end{cases}$$

$$l_{ij}=d_{ij}-w_{ij}$$

定义二分图的切图权重

$$W(A_k,\bar{A}_k)=\sum_{i\in A_k,j\notin A_k}w_{ij} \tag{5.8}$$

对于图 G 的一种切割 $(A_1,...,A_d)$，定义 NCut 权重为

$$\mathrm{NCut}(A_1,...,A_d)=\sum_{i=1}^{d}\frac{W(A_i,\bar{A}_i)}{\mathrm{vol}(A_i)} \tag{5.9}$$

其中 $\mathrm{vol}(A_i)$ 表示第 i 子图内节点到所有节点连接边的权重之和

$$\mathrm{vol}(A_i)=\sum_{j\in A_i,k\in V}w_{jk}=\sum_{j\in A_i}d_{jj} \tag{5.10}$$

对于图 G 各节点分配到子图的情况，我们定义**指示矩阵** $H=[\mathbf{h}_1,\mathbf{h}_2,...,\mathbf{h}_d]\in R^{N\times d}$

$$h_{ji}=\begin{cases}0 & v_j\notin A_i\\ \frac{1}{\sqrt{\mathrm{vol}(A_j)}} & v_j\in A_i\end{cases} \tag{5.11}$$

选取指示矩阵列向量 $\mathbf{h}_i$，计算其关于拉普拉斯矩阵的二次型，有

$$\begin{aligned}\mathbf{h}_i^T L\mathbf{h}_i =& \sum_{j=1}^{N}\sum_{k=1}^{N}(d_{jk}-w_{jk})h_{ji}h_{ki}\\ =& \sum_{j=1}^{N} h_{jj}^2 d_{jj} - \sum_{j=1}^{N}\sum_{k\neq j} w_{jk}h_{ji}h_{ki}\\ =& \sum_{j=1}^{N} h_{jj}^2 \sum_{k=1}^{N} w_{jk} - \sum_{j=1}^{N}\sum_{k\neq j} w_{jk}h_{ji}h_{ki}\\ =& \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} w_{jk}(h_{ji}-h_{ki})^2\end{aligned}$$

由于 H 只有两种取值，只有 $h_{ji}-h_{ki}$ 不等，即 v_j 和 v_k 不同时属于 A_i 时，才会对整体有贡献

$$\begin{aligned}\mathbf{h}_i^T L\mathbf{h}_i =& \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} w_{jk}(h_{ji}-h_{ki})^2\\ =& \frac{1}{2}\sum_{j\in A_i,k\notin A_i} w_{jk}(h_{ji}-h_{ki})^2 + \frac{1}{2}\sum_{j\notin A_i,k\in A_i} w_{jk}(h_{ji}-h_{ki})^2\\ =& \frac{1}{2}\sum_{j\in A_i,k\notin A_i} w_{jk}\left(\frac{1}{\sqrt{\mathrm{vol}(A_i)}}-0\right)^2 + \frac{1}{2}\sum_{j\notin A_i,k\in A_i} w_{jk}\left(0-\frac{1}{\sqrt{\mathrm{vol}(A_i)}}\right)^2\\ =& \frac{1}{2}W(A_i,\bar{A}_i)\frac{1}{\mathrm{vol}(A_i)} + \frac{1}{2}W(A_i,\bar{A}_i)\frac{1}{\mathrm{vol}(A_i)}\\ =& \frac{W(A_i,\bar{A}_i)}{\mathrm{vol}(A_i)}\end{aligned}$$

因此

$$\mathrm{tr}(H^T LH) = \sum_{i=1}^{d}\frac{W(A_i,\bar{A}_i)}{\mathrm{vol}(A_i)} = \mathrm{NCut}(A_1,...,A_d) \tag{5.12}$$

即 NCut 的优化问题可以表示为一个简单的二次型优化问题。

优化问题中，为了防止 H 退化为平凡解，需要添加约束。根据 H 的定义，有

$$\begin{aligned}\mathbf{h}_i^T D\mathbf{h}_i = \mathbf{h}_i^T D\mathbf{h}_i =& \sum_{j=1}^{N}\sum_{k=1}^{N} h_{ji}d_{jk}h_{ki}\\ =& \sum_{j=1}^{N} h_{ji}^2 d_{jj} = \frac{1}{\mathrm{vol}(A_i)}\sum_{j\in A_i} d_{jj}\\ =& \frac{1}{\mathrm{vol}(A_i)}\mathrm{vol}(A_i) = 1\end{aligned}$$

且 H 列向量之间正交，因此

$$H^T DH = I$$

所以优化问题为

$$\begin{aligned}\min_{H}\quad & J(H) = \mathrm{tr}(H^T LH)\\ \text{s.t.}\quad & H^T DH = I\end{aligned} \tag{5.13}$$

该形式和 LE 中的优化问题，即式4.10完全一样，此处的 H 相当于 LE 中的 Y^T。使用 Lagrange 乘子法

$$L(H,\lambda)=\mathrm{tr}(H^TLH)+\lambda(\mathrm{tr}(H^TDH)-2d)$$

求偏导

$$\frac{\partial L}{\partial H}=2LH-2\lambda DH=0$$

$$LH=\lambda DH$$

$$D^{-1}LH=\lambda H$$

即 H 的列向量为 $D^{-1}L$ 矩阵的列特征向量。取最小的 d 个特征值对应的特征向量组成 H 即可。

由于没有限制 H 的取值范围，因此计算出的 H 不一定满足前面的定义。因此可以将它的行作为各样本的降维特征进行简单聚类，就可以得到聚类结果。

5.3 自组织映射 SOM

SOM 是一种用于特征检测的无监督学习神经网络，除了可以用于聚类外，也可用于数据降维

算法	**自组织映射**
算法简述	对于聚类问题，根据最佳匹配单元迭代调整聚类中心位置
已知	样本 $X\in R^{n\times N}$，超参数 σ，类别数 K，递减学习率函数 $\alpha(t)$
求	分配矩阵 $R\in\{0,1\}^{N\times K}$ 和分配中心 $\mathbf{c}_1,\mathbf{c}_2,...,\mathbf{c}_K$
解类型	迭代解
算法步骤	1. 随机初始化 K 个分配中心 2. 从样本集随机挑选样本 $\mathbf{x}(t)$，计算和每个中心之间的距离 $d(\mathbf{c}_i(t),t)=\|\|\mathbf{x}(t)-\mathbf{c}_i(t)\|\|$ 3. 选出距离最小的中心为最佳匹配单元 $\mathbf{c}_{BMU}(t)=\arg\min_c d(\mathbf{c},t)$ 4. 各分配中心按距离最佳匹配单元距离调整自身位置 $\mathbf{c}_i(t+1)=\mathbf{c}_i(t)+\alpha(t)\exp\left\{-\frac{\|\|\mathbf{c}_i(t)-\mathbf{c}_{BMU}(t)\|\|^2}{\sigma^2}\right\}(\mathbf{x}(t)-\mathbf{c}_i(t))$ 5. 若分配中心收敛，停止迭代，否则回到 2 6. 将距离每个样本最近的分配中心作为分配结果

Part III

概率方法

6 概率求解

概率机器学习是一大类机器学习方法的总称。它们通过含参数的概率模型对变量分布进行建模。频率学派希望求解参数点估计，并使用某种最优参数进行预测。贝叶斯学派希望求出参数的分布（即贝叶斯推断），并在此基础上求出样本的生成、判别或预测模型（即贝叶斯决策）。它们的共同点是：使用含参数的概率模型。

然而，有时建模的概率分布的表达形式会相当复杂，以至于无法解析求解。这时，就需要一些迭代或者近似算法进行求解。本章不针对某一个具体问题、具体模型或具体算法，而是介绍通用的概率分布的迭代或近似算法。在下一章中将会展示这些通用算法如何服务于具体模型和问题。

6.1 基本任务

对于概率机器学习模型，有三个主要任务，分别是参数估计任务、分布估计任务和分布采样任务。本章介绍的算法都是为了解决这三大任务来设计的。

参数估计（点估计）任务：给定数据集 X，给定未知参数 θ 的概率模型，求出使得某一种指标最优的参数 $\hat{\theta}$。指标往往选取最大后验（MAP）、最大似然（MLE）或最小化某两个分布间的 KL 散度等。

分布估计任务：给定数据集 X，给定未知参数 θ 的概率模型，求出参数 θ 在数据集条件下的后验分布 $p(\theta|X)$。

分布采样任务：给定已知参数 θ 的概率模型 $p(\mathbf{x}|\theta)$，求一组采样样本 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)} \sim p(\mathbf{x}|\theta)$。

这些任务和基本任务的关系是：通过对基本任务进行假设定义概率模型，就可以把基本任务转化为这些概率机器学习任务。而这些概率机器学习任务又有一些通用的解决方法。如 E-M 算法之于点估计任务，MCMC 之于采样任务。对于具体的概率机器学习算法，就会将通用解决方法应用到具体模型上来。

这里还要特别解释一下分布采样任务。采样并不和任何机器学习基本任务直接相关，但采样往往是参数估计或分布估计求解过程中的必要步骤。因此，我们选择将其列入概率机器学习的主要任务之一。

6.2 E-M 算法

在概率机器学习方法中，有一大类模型会涉及不完全观测问题。即：有无法得知真实值的随机变量影响数据的分布。观察到的数据分布 $p(\mathbf{x})$，实际上是观测变量和这些变量 $\mathbf{z}$ 的联合分布的积分

$$p(\mathbf{x}) = p(\mathbf{x}, \mathbf{z})d\mathbf{z}$$

这种无法观测真实值的变量称为**隐变量**。隐变量有多种来源，例如：含有噪声的数据的实际值、数据来自多个分布中的哪一个分布、数据在低维流形上的嵌入等等。

然而，如果能对隐变量对观测变量影响的形式进行建模，或者说如果能得到隐变量和观测变量联合分布的形式，而允许其中的参数未知，即

$$p(\mathbf{x}|\theta) = p(\mathbf{x}, \mathbf{z}|\theta)d\mathbf{z}$$

那么就可以设计算法，仅仅根据观测到的数据同时推理出未知参数和隐变量分布。这就是 E-M 算法。

算法	**E-M 算法**
算法简述	对于观测不完全的参数 MLE 估计问题，交替得到似然下界和更优参数值
已知	观测变量 $X \in R^{n\times N}$，联合分布形式 $p(\mathbf{x},\mathbf{z}\|\theta)$ ，隐变量后验分布形式 $p(\mathbf{z}\|\mathbf{x},\theta)$
求	最优参数 $\theta^\star$ 使得最大化观测似然 $\max_\theta \log L(\theta) = \sum_{i=1}^{N} \log \int p(\mathbf{x}^{(i)},\mathbf{z}\|\theta) d\mathbf{z}$
解类型	迭代解
算法步骤	1. 初始化参数 θ 2. E 步：求出**联合分布对数**（即似然下界）对隐变量后验的期望 $LB(\theta,\theta(t)) = E_{z\sim p(z\|x,\theta(t))}[\log p(\mathbf{x},\mathbf{z}\|\theta)]$ 3. M 步：最大化下界函数，求出新的参数值 $\theta(t+1) = \arg\max_\theta LB(\theta,\theta(t))$ 4. 计算 $\log L(\theta(t+1))$，若收敛则停止，否则回第 2 步

6.2.1 核心结论证明

E-M 算法的核心在于这样的一个结论

$$\log p(\mathbf{x},\mathbf{z}|\theta)d\mathbf{z} \geq \int \log \frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})} q(\mathbf{z})d\mathbf{z},\ \forall\theta,\ \forall \int q(\mathbf{z})d\mathbf{z} = 1 \tag{6.1}$$

下面我们利用两种方法对其进行证明。

证明. **Jensen 不等式证明**

假设有任意关于隐变量的概率分布 $q(\mathbf{z})$ 满足

$$\int q(\mathbf{z})d\mathbf{z} = 1$$

有恒等式

$$p(\mathbf{x}|\theta) = \int q(\mathbf{z}) \frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})} d\mathbf{z},\ \forall\theta, q$$

两边取对数，即得对数似然

$$\log L(\theta) = \log \int q(\mathbf{z}) \frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})} d\mathbf{z},\ \forall\theta, q$$

等号右侧可以写成期望的形式

$$\log L(\theta) = \log E_{z\sim q}[\frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})}],\ \forall\theta, q$$

由 Jensen 不等式，期望的对数大于等于对数的期望

$$\log L(\theta) \geq E_{z\sim q}[\log \frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})}],\ \forall\theta, q \tag{6.2}$$

注意该式对于任意 θ 都成立。因此，不等号右侧的部分是左侧对数似然的一个下界，我们称之为“证据下界”（Evidence Lower Bound, ELBO）。

根据 Jensen 不等式性质，取等条件为期望内的变量为常数

$$\frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})}=C,\ \forall\mathbf{z} \tag{6.3}$$

即

$$q(\mathbf{z})=\frac{1}{C}p(\mathbf{x},\mathbf{z}|\theta),\ \forall\mathbf{z}$$

两边对 $\mathbf{z}$ 积分

$$1=\int\frac{1}{C}p(\mathbf{x},\mathbf{z}|\theta)d\mathbf{z}$$

即

$$C=p(\mathbf{x}|\theta)$$

代入式6.3中，有

$$q(\mathbf{z})=\frac{p(\mathbf{x},\mathbf{z}|\theta)}{p(\mathbf{x}|\theta)}=p(\mathbf{x}|\mathbf{z},\theta),\ \forall\mathbf{z}$$

也就是说，式6.2取等条件为，隐变量分布和隐变量对观测变量以及参数的条件分布相同。
ELBO 下界还可以继续优化

$$\log L(\theta)\geq E_{z\sim q}[\log p(\mathbf{x},\mathbf{z}|\theta)]+\int -q(\mathbf{z})\log q(\mathbf{z})d\mathbf{z},\ \forall\theta,q$$

后一项为分布 $q(\mathbf{z})$ 的香农熵，是一个非负值，因此可以得到一个似然下界

$$\log L(\theta)\geq E_{z\sim q}[\log p(\mathbf{x},\mathbf{z}|\theta)],\ \forall\theta,q$$

□

证明. **KL 散度证明**

对于边缘概率，我们有贝叶斯公式

$$p(\mathbf{x}|\theta)=\frac{p(\mathbf{x},\mathbf{z}|\theta)}{p(\mathbf{z}|\mathbf{x},\theta)},\ \forall\mathbf{z},\theta$$

两边取对数

$$\log p(\mathbf{x}|\theta)=\log p(\mathbf{x},\mathbf{z}|\theta)-\log p(\mathbf{z}|\mathbf{x},\theta),\ \forall\mathbf{z},\theta$$

假设有任意关于隐变量的概率分布 $q(\mathbf{z})$ 满足

$$\int q(\mathbf{z})d\mathbf{z}=1$$

有恒等式

$$\log p(\mathbf{x}|\theta)=\log\frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})}-\log\frac{p(\mathbf{z}|\mathbf{x},\theta)}{q(\mathbf{z})},\ \forall\mathbf{z},\theta,q$$

这一步用了加一项减一项的技巧。两边求期望

$$E_{z\sim q}[\log p(\mathbf{x}|\theta)] = E_{z\sim q}[\log \frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})}] + E_{z\sim q}[-\log \frac{p(\mathbf{z}|\mathbf{x},\theta)}{q(\mathbf{z})}],\ \forall\theta, q$$

左侧和 $\mathbf{z}$ 无关，可以去掉期望。右侧展开为积分形式

$$\log p(\mathbf{x}|\theta) = \log L(\theta) = \int q(\mathbf{z}) \log \frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})} d\mathbf{z} + \int -q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x},\theta)}{q(\mathbf{z})}] d\mathbf{z},\ \forall\theta, q$$

等号右侧第二项实际上就是两个分布之间的 KL 散度，它可以衡量两个分布之间的距离（然而它并不对称，不能作为距离），具有非负性质

$$\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x},\theta)] = \int -q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x},\theta)}{q(\mathbf{z})}] d\mathbf{z} \geq 0,\ \forall\theta, q$$

因此，另一项就是对数似然的下界，称为证据下界 ELBO

$$\log L(\theta) \geq \int q(\mathbf{z}) \log \frac{p(\mathbf{x},\mathbf{z}|\theta)}{q(\mathbf{z})} d\mathbf{z},\ \forall\theta, q$$

而不等号取等的条件也非常明显，就是 KL 散度 $\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x},\theta)] = 0$，即 $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x},\theta)$

此时，ELBO 还可以继续化简

$$\begin{aligned}
\log L(\theta) &\geq \int q(\mathbf{z}) \log p(\mathbf{x},\mathbf{z}|\theta) d\mathbf{z} - \int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z},\ \forall\theta \\
&= \int q(\mathbf{z}) \log p(\mathbf{x},\mathbf{z}|\theta) d\mathbf{z} + \mathrm{H}[q(\mathbf{z})] \\
&\geq \int q(\mathbf{z}) \log p(\mathbf{x},\mathbf{z}|\theta) d\mathbf{z},\ \forall\theta, q
\end{aligned}$$

右侧的香农熵非负，且与 θ 无关。

□

6.2.2 迭代算法

上面我们通过两种方式证明了结论6.1，并推理出了取等条件 $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x},\theta)$。那么，为了求出使对数似然最大的 θ，我们可以采取这样的方法：

首先，设定初始参数值 $\theta(0)$。其次，求出隐变量后验分布 $p(\mathbf{z}|\mathbf{x},\theta(0))$。令分布 $q(\mathbf{z})$ 等于该分布，此时 ELBO 是一个关于 θ 的函数

$$\mathrm{ELBO}(\theta|\theta(0)) = \int \log \frac{p(\mathbf{x},\mathbf{z}|\theta)}{p(\mathbf{z}|\mathbf{x},\theta(0))} p(\mathbf{z}|\mathbf{x},\theta(0)) d\mathbf{z}$$

这个函数的特点是：在 $\theta = \theta(0)$ 时，该函数和对数似然相等。而在其他时候，对数似然一定大于该函数。因此，如果可以找到使 ELBO 更优的参数值

$$\theta(1) = \arg\max_{\theta} \mathrm{ELBO}(\theta|\theta(0))$$

那么对应的对数似然也一定比当前的对数似然更优，即

$$\log L(\theta(0)) = \mathrm{ELBO}(\theta(0)|\theta(0)) \leq \mathrm{ELBO}(\theta(1)|\theta(0)) \leq \log L(\theta(1))$$

由于 ELBO 中含有和 θ 无关的 $\mathrm{H}[q]$ 一项，将其删去不影响优化结果。因此实际上我们会使用

$$\theta(t) = \arg\max_{\theta} \int p(\mathbf{z}|\mathbf{x}, \theta(t-1)) \log p(\mathbf{x}, \mathbf{z}|\theta) d\mathbf{z}$$

这样，我们就可以迭代这一过程，直到 $\log L(\theta)$ 收敛。这就是 E-M 算法。总结一下，每一轮迭代中，我们分别要：

E 步（Estimation）：求出对数似然的下界函数，即**对数联合分布**对隐变量条件分布的期望

$$LB(\theta, \theta(t)) = E_{z\sim p(z|x,\theta(0))}[\log p(\mathbf{x}, \mathbf{z}|\theta)] \tag{6.4}$$

M 步（Maximization）：最大化下界函数，求出新的参数值

$$\theta(t+1) = \arg\max_{\theta} LB(\theta, \theta(t)) \tag{6.5}$$

注：E-M 算法也可以用来求最大后验 MAP 解，只需要在参数更新一步的目标函数中加一项对数先验 $p(\theta)$ 即可。

6.2.3 ELBO 的不同形式

我们可以得到不同形式的 ELBO 表达式

$$\begin{aligned}
\text{ELBO} &= \left\langle \log \frac{p(\mathbf{x}, \mathbf{z}|\theta)}{q(\mathbf{z}|\mathbf{x}, \phi)} \right\rangle_{q(\mathbf{z}|\mathbf{x},\phi)} \\
&= \langle \log p(\mathbf{x}, \mathbf{z}|\theta) \rangle_{q(\mathbf{z}|\mathbf{x},\phi)} + \mathrm{H}[q(\mathbf{z}|\mathbf{x}, \phi)] \\
&= \langle \log p(\mathbf{x}|\mathbf{z}, \theta) \rangle_{q(\mathbf{z}|\mathbf{x},\phi)} - \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\theta)]
\end{aligned} \tag{6.6}$$

证明. **第一二种形式**

$$\begin{aligned}
\log L(\theta) &= \log p(\mathbf{x}) = \int \log p(\mathbf{x}) q(\mathbf{z}|\mathbf{x}, \phi) d\mathbf{z} \\
&= \int \log \frac{p(\mathbf{x}, \mathbf{z}|\theta) q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z}|\mathbf{x}, \theta) q(\mathbf{z}|\mathbf{x}, \phi)} q(\mathbf{z}|\mathbf{x}, \phi) d\mathbf{z} \\
&= \left\langle \log p(\mathbf{x}, \mathbf{z}|\theta) - \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z}|\mathbf{x}, \theta)} \right\rangle_{q(\mathbf{z}|\mathbf{x},\phi)} \\
&= \langle \log p(\mathbf{x}, \mathbf{z}|\theta) \rangle_{q(\mathbf{z}|\mathbf{x},\phi)} + \mathrm{H}[q(\mathbf{z}|\mathbf{x}, \phi)] + \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \theta)] \\
&= \text{ELBO} + \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \theta)]
\end{aligned}$$

第三种形式

$$\begin{aligned}
\log L(\theta) &= \log p(\mathbf{x}) = \int \log p(\mathbf{x}) q(\mathbf{z}|\mathbf{x}, \phi) d\mathbf{z} \\
&= \int \log \frac{p(\mathbf{x}|\mathbf{z}, \theta) p(\mathbf{z}|\theta) q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z}|\mathbf{x}, \theta) q(\mathbf{z}|\mathbf{x}, \phi)} q(\mathbf{z}|\mathbf{x}, \phi) d\mathbf{z} \\
&= \left\langle \log p(\mathbf{x}|\mathbf{z}, \theta) + \log \frac{p(\mathbf{z}|\theta)}{q(\mathbf{z}|\mathbf{x}, \phi)} + \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z}|\mathbf{x}, \theta)} \right\rangle_{q(\mathbf{z}|\mathbf{x},\phi)} \\
&= \langle \log p(\mathbf{x}|\mathbf{z}, \theta) \rangle_{q(\mathbf{z}|\mathbf{x},\phi)} - \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\theta)] + \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \theta)] \\
&= \text{ELBO} + \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \theta)]
\end{aligned}$$

□

6.3 近似方法

6.3.1 变分近似

E-M 算法解决了包含隐变量的分布参数估计问题，但无法给出参数的分布。对于隐变量的分布也只能给出参数为极大似然解下的情况。

然而，如果要求解参数的分布 $p(\theta|\mathbf{x})$，就需要对参数、隐变量和样本联合分布 $p(\theta,\mathbf{z},\mathbf{x})$ 进行积分。一方面，要对隐变量积分。另一方面，为了求解条件概率，对样本也要积分。这个积分在通常情况下无法解析计算，采样计算成本也很高。

变分推断方法的解决方案是：使用一组较为简单的分布 $q(\mathbf{z})$ 近似 $p(\mathbf{z}|\mathbf{x})$，并求出最优的近似。这里原模型参数 θ 并入了隐变量 $\mathbf{z}$ 中。这里，可以将优化目标用 KL 散度表示

$$\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})] = \int q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}$$

根据式6.6，有

$$\log p(\mathbf{x}) = \int q(\mathbf{z}) \log p(\mathbf{x},\mathbf{z}) d\mathbf{z} + \mathrm{H}[q(\mathbf{z})] + \mathrm{KL}[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})]$$

即对数边缘分布等于近似 KL 散度和 ELBO 之和。由于等号左边的 $\log p(\mathbf{x})$ 和 $q(\mathbf{z})$ 无关，因此最小化 KL 散度就相当于最大化 ELBO。即

$$\max_q L[q] = \int q(\mathbf{z}) \log p(\mathbf{x},\mathbf{z}) d\mathbf{z} + \mathrm{H}[q(\mathbf{z})] \tag{6.7}$$

坐标下降变分推断 CAVI

算法	**坐标下降变分推断**
算法简述	对于隐变量分布求解问题，使用平均场假设分解隐变量联合分布， 利用变分最优解条件对隐变量后验分布的参数进行迭代优化
已知	联合分布 $p(\mathbf{x},\mathbf{z})$，近似分布形式 $q(\mathbf{z}\|\phi)$
求	隐变量后验分布 $p(\mathbf{z}\|\mathbf{x})$ 的最优近似 $q(\mathbf{z}\|\phi)$
解类型	近似解
近似解	1. 将隐变量划分为多个子集 $\mathbf{z}_1,\mathbf{z}_2,...,\mathbf{z}_K$，从而 $q(\mathbf{z}\|\phi)=\prod_{i=1}^K q_i(\mathbf{z}_i\|\phi_i)$ 2. 根据最优条件 $\log q_i(\mathbf{z}_i\|\phi_i) = \int q_{-i}(\mathbf{z}_{-i}\|\phi_{-i}) \log p(\mathbf{x},\mathbf{z}) d\mathbf{z}_{-i} + const$， 得到隐变量间互相依赖的表达式 $\phi_i = E[f_i(\mathbf{z}_{-i},\mathbf{x})]$ 3. 任意初始化 ϕ，循环更新 $\mathbf{z}_i$，直到收敛。 $\phi_i(t+1) = E[f_i(\mathbf{z}_1(t+1),...,\mathbf{z}_{i-1}(t+1),\mathbf{z}_{i+1}(t),...,\mathbf{z}_K(t),\mathbf{x})]$ $\mathbf{z}_i(t+1) = \int \mathbf{z}_i q(\mathbf{z}_i\|\phi_i(t+1)) d\mathbf{z}_i$

上述推导中，优化变量是一个函数，q 而不是模型参数 θ，并且有分布约束

$$\int q(\mathbf{z}) d\mathbf{z} = 1$$

也就是说，这是一个典型的泛函极值问题。直接求解该问题很麻烦，因此我们考虑引入独立性假设简化计算。根据平均场理论，假设 $\mathbf{z}_1,\mathbf{z}_2,...,\mathbf{z}_K$ 是隐变量 $\mathbf{z}$ 的一种划分，在这种划分下联合分布等于各自分布的乘积

$$q(\mathbf{z}) = \prod_{i=1}^{K} q_i(\mathbf{z}_i) \tag{6.8}$$

注意此处划分后的 $\mathbf{z}_i$ 不必须为一维变量，可以是一组联合分布很好求的隐变量。为简便起见，我们记 $\mathbf{z}$ 除去 $\mathbf{z}_i$ 的部分为 $\mathbf{z}_{-i}$。这一，就可以化简 $L[q]$。其第一项为

$$\begin{aligned}
&\int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \\
=&\int q_i(\mathbf{z}_i) \int q_{-i}(\mathbf{z}_{-i}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{-i} d\mathbf{z}_i \\
=&\int q_i(\mathbf{z}_i) \big\langle \log p(\mathbf{x}, \mathbf{z}) \big\rangle_{q_{-i}(\mathbf{z}_{-i})} d\mathbf{z}_i
\end{aligned}$$

定义

$$q_i^\star(\mathbf{z}_i) = \frac{1}{C} \exp\{\big\langle \log p(\mathbf{x}, \mathbf{z}) \big\rangle_{q_{-i}(\mathbf{z}_{-i})}\}$$

其中 C 为 $q_i^\star(\mathbf{z}_i)$ 的归一化常数，于是有

$$\int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int q_i(\mathbf{z}_i) \log q_i^\star(\mathbf{z}_i) d\mathbf{z}_i + const$$

同理，另一项

$$\begin{aligned}
\mathrm{H}[q(\mathbf{z})] =& -\int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z} \\
=& -\int \prod_{i=1}^{K} q_i(\mathbf{z}_i) (\sum_{k=1}^{K} \log q_k(\mathbf{z}_k)) d\mathbf{z} \\
=& -\sum_{k=1}^{K} \int \int \prod_{i=1}^{K} q_i(\mathbf{z}_i) \log q_k(\mathbf{z}_k) d\mathbf{z}_{-k} d\mathbf{z}_k \\
=& -\sum_{k=1}^{K} \int q_k(\mathbf{z}_k) \log q_k(\mathbf{z}_k) d\mathbf{z}_k
\end{aligned}$$

因此有

$$\begin{aligned}
L[q] =& \int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} + \mathrm{H}[q(\mathbf{z})] \\
=& \int q_i(\mathbf{z}_i) \log q_i^\star(\mathbf{z}_i) d\mathbf{z}_i - \sum_{k=1}^{K} \int q_k(\mathbf{z}_k) \log q_k(\mathbf{z}_k) d\mathbf{z}_k + const \\
=& \int q_i(\mathbf{z}_i) (\log q_i^\star(\mathbf{z}_i) - log q_i(\mathbf{z}_i)) d\mathbf{z}_i - \sum_{k \neq i}^{K} \int \log q_k(\mathbf{z}_k) d\mathbf{z}_k + const \\
=& -\mathrm{KL}[q_i(\mathbf{z}_i) || q_i^\star(\mathbf{z}_i)] + \mathrm{H}[q_{-i}(\mathbf{z}_{-i})] + const
\end{aligned}$$

由于 KL 散度有非负性，因此当 KL 散度为 0 时，泛函取最大值，同时约束条件可以满足，即

$$q_i(\mathbf{z}_i) = q_i^\star(\mathbf{z}_i) = \frac{\exp\{\big\langle \log p(\mathbf{x}, \mathbf{z}) \big\rangle_{q_{-i}(\mathbf{z}_{-i})}\}}{\int \exp\{\big\langle \log p(\mathbf{x}, \mathbf{z}) \big\rangle_{q_{-i}(\mathbf{z}_{-i})}\} d\mathbf{z}_i}, \forall i \tag{6.9}$$

对式 6.9两边取对数，有

$$\log q_i(\mathbf{z}_i) = \left\langle \log p(\mathbf{x}, \mathbf{z}) \right\rangle_{q_{-i}(\mathbf{z}_{-i})} + const \tag{6.10}$$

这样，我们就得到了变分推断最优分布满足的条件。

实际使用中，如果已知 $\mathbf{z}$ 似然函数的共轭分布，就能得到后验分布的具体形式。此时后验分布就会变成未知参数的分布 $q(\mathbf{z}|\phi)$。因此有

$$\log q_i(\mathbf{z}_i|\phi_i) = \int q_{-i}(\mathbf{z}_{-i}|\phi_{-i}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{-i} + const \tag{6.11}$$

由于一般遇到的都是指数族分布，根据式6.11，我们可以得到 ϕ_i 关于 $\mathbf{z}_{-i}$ 的 n 阶矩的表达式，也就是得到最优参数之间循环依赖的关系

$$\phi_i = E[f_i(\mathbf{z}_{-i}, \mathbf{x})]$$

其中期望是隐变量对其后验分布取期望，和 $\mathbf{x}$ 无关。类似于强化学习值迭代以及 Gibbs 采样，求解循环依赖的一种简单的思路是循环更新每一组划分。这样的算法也称为坐标下降算法。

$$\phi_i(t+1) = E[f_i(\mathbf{z}_1(t+1), ..., \mathbf{z}_{i-1}(t+1), \mathbf{z}_{i+1}(t), ..., \mathbf{z}_K(t), \mathbf{x})] \tag{6.12}$$

其中

$$\mathbf{z}_i(t) = \int \mathbf{z}_i q(\mathbf{z}_i|\phi_i(t)) d\mathbf{z}_i$$

随机梯度变分推断 SGVI

坐标下降变分推断需要先验地将参数进行划分，划分后不同划分之间假设独立。这就限制了模型的表达能力，也使近似分布不够精确。前面提到过，梯度下降是一种可导优化问题的通用迭代算法。如果将优化目标用参数形式表示，求出优化目标对参数导数，就可以直接迭代更新。

算法	**随机梯度变分推断**
算法简述	对于隐变量分布求解问题，使用平均场假设分解隐变量联合分布， 利用变分最优解条件对隐变量后验分布的参数进行迭代优化
已知	联合分布 $p(\mathbf{x}, \mathbf{z})$，近似分布形式 $q(\mathbf{z}|\phi)$ 隐变量重参数化 $\mathbf{z} = g(\epsilon, \mathbf{x}, \phi)$，$\epsilon \sim P(\epsilon)$
求	隐变量后验分布 $p(\mathbf{z}|\mathbf{x})$ 的最优近似 $q(\mathbf{z}|\phi)$
解类型	近似解
近似解	1. 从 $P(\epsilon)$ 中采样 M 个噪声 $\epsilon^{(1)}, \epsilon^{(2)}, ..., \epsilon^{(M)}$ 2. 随机从样本集中选取样本 $\mathbf{x}$，计算梯度 $\nabla_{\phi(t)} L(\phi)(t) \approx \frac{1}{M} \sum_{i=1}^{M} \nabla_{z^{(i)}} (\log q(\mathbf{z}^{(i)}|\phi(t)) - \log p(\mathbf{x}, \mathbf{z}^{(i)})) \nabla_{\phi(t)} \mathbf{z}^{(i)}$， 其中 $\nabla_{\phi(t)} \mathbf{z}^{(i)} = \nabla_{\phi(t)} g(\epsilon^{(i)}, \mathbf{x}, \phi(t))$ 3. 进行梯度下降 $\phi(t+1) = \phi(t) - \alpha \nabla_{\phi(t)} L(\phi)(t)$

具体来说，假设已知参数联合分布 $p(\mathbf{x}, \mathbf{z})$，并且有了隐变量后验分布的参数化近似形式 $q(\mathbf{z}|\phi)$，那么优化问题就转变为关于参数 ϕ 的优化问题

$$\max_{\phi} L(\phi) = \int q(\mathbf{z}|\phi)(\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\phi)) d\mathbf{z}$$

这个问题是最大化问题。为了和前面梯度下降的表达一致，我们对目标函数取反变成最小化问题

$$\min_{\phi} L(\phi) = \int q(\mathbf{z}|\phi)(\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))d\mathbf{z}$$

求梯度

$$\begin{aligned}\nabla_\phi L(\phi) =& \nabla_\phi \int q(\mathbf{z}|\phi)(\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))d\mathbf{z} \\ =& \int (\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))\nabla_\phi q(\mathbf{z}|\phi) + q(\mathbf{z}|\phi)\nabla_\phi(\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))d\mathbf{z}\end{aligned}$$

梯度分成了两项，先考虑右侧一项

$$\begin{aligned}&\int q(\mathbf{z}|\phi)\nabla_\phi(\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))d\mathbf{z} \\ =& \int q(\mathbf{z}|\phi)\nabla_\phi \log q(\mathbf{z}|\phi)d\mathbf{z} \\ =& \int q(\mathbf{z}|\phi)\frac{1}{q(\mathbf{z}|\phi)}\nabla_\phi q(\mathbf{z}|\phi)d\mathbf{z} \\ =& \nabla_\phi \int q(\mathbf{z}|\phi)d\mathbf{z} \\ =& \nabla_\phi 1 = 0\end{aligned}$$

因此梯度就等于左侧的项

$$\begin{aligned}\nabla_\phi L(\phi) =& \int (\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))\nabla_\phi q(\mathbf{z}|\phi)d\mathbf{z} \\ =& \int q(\mathbf{z}|\phi)(\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))\frac{\nabla_\phi q(\mathbf{z}|\phi)}{q(\mathbf{z}|\phi)}d\mathbf{z} \\ =& \int q(\mathbf{z}|\phi)(\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))\nabla_\phi \log q(\mathbf{z}|\phi)d\mathbf{z} \\ =& \left\langle (\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))\nabla_\phi \log q(\mathbf{z}|\phi) \right\rangle_{q(\mathbf{z}|\phi)}\end{aligned}$$

这个期望可以使用本章第 3 节介绍的采样方法进行计算。首先从 $q(\mathbf{z}|\phi(t))$ 中采样 M 个隐变量 $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(M)}$，接下来计算

$$\begin{aligned}\nabla_{\phi(t)} L(\phi)(t) =& \left\langle (\log q(\mathbf{z}|\phi(t)) - \log p(\mathbf{x}, \mathbf{z}))\nabla_{\phi(t)} \log q(\mathbf{z}|\phi(t)) \right\rangle_{q(\mathbf{z}|\phi(t))} \\ \approx& \frac{1}{M}\sum_{i=1}^{M}(\log q(\mathbf{z}^{(i)}|\phi(t)) - \log p(\mathbf{x}, \mathbf{z}^{(i)}))\nabla_{\phi(t)} \log q(\mathbf{z}^{(i)}|\phi(t))\end{aligned}$$

$$\phi(t+1) = \phi(t) - \alpha\nabla_{\phi(t)} L(\phi)(t)$$

其中 α 是学习率。

不过这样的方法存在一定的问题，因为

$$\nabla_{\phi(t)} \log q(\mathbf{z}^{(i)}|\phi(t)) = \frac{\nabla_{\phi(t)} q(\mathbf{z}^{(i)}|\phi(t))}{q(\mathbf{z}^{(i)}|\phi(t))}$$

如果采样到的 $\mathbf{z}^{(i)}$ 对应的概率密度很小，梯度绝对值就会很大，从而整个梯度项在不同采样之间的方差会很大，不利于随机梯度算法收敛。因此我们需要避免 log 的出现。

一种解决方法是使用**重参数化技巧**，将采样产生的非确定性变量变为确定性变量，从而使其可以进行梯度计算。具体来说，我们假设 $\mathbf{z}$ 是由来自另一简单分布的随机变量 $\epsilon \sim P(\epsilon)$ 通过确定函数计算得到的

$$\mathbf{z} = g(\epsilon, \mathbf{x}, \phi)$$

例如 $\mathbf{z}$ 满足均值为 μ_z，方差为 Σ_z 的高斯分布时，可以令 $P(\epsilon) = N(\epsilon; \mathbf{0}, I)$，从而有 $\mathbf{z} = \mu + \Sigma\epsilon$。这样，就可以计算采样变量对于采样参数的偏导 $\frac{\partial \mathbf{z}}{\partial \phi}$。由于

$$\int q(\mathbf{z}|\phi)d\mathbf{z} = 1 = \int P(\epsilon)d\epsilon$$

两边求全微分，有

$$q(\mathbf{z}|\phi)d\mathbf{z} = P(\epsilon)d\epsilon$$

因此损失函数梯度

$$\begin{aligned}
&\nabla_\phi L(\phi) = \nabla_\phi \int (\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))q(\mathbf{z}|\phi)d\mathbf{z} \\
=&\int \nabla_z (\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))\nabla_\phi g(\epsilon, \mathbf{x}, \phi)P(\epsilon)d\epsilon \\
=&\left\langle \nabla_z (\log q(\mathbf{z}|\phi) - \log p(\mathbf{x}, \mathbf{z}))\nabla_\phi g(\epsilon, \mathbf{x}, \phi) \right\rangle_{P(\epsilon)}
\end{aligned}$$

最终的迭代算法如下。首先从 $P(\epsilon)$ 中采样 M 个噪声 $\epsilon^{(1)}, \epsilon^{(2)}, ..., \epsilon^{(M)}$，接下来计算

$$\nabla_{\phi(t)} L(\phi)(t) \approx \frac{1}{M} \sum_{i=1}^{M} \nabla_{z^{(i)}} (\log q(\mathbf{z}^{(i)}|\phi(t)) - \log p(\mathbf{x}, \mathbf{z}^{(i)}))\nabla_{\phi(t)} g(\epsilon^{(i)}, \mathbf{x}, \phi(t)) \tag{6.13}$$

其中

$$\mathbf{z}^{(i)} = g(\epsilon^{(i)}, \mathbf{x}, \phi(t)) \tag{6.14}$$

设 α 是学习率，有

$$\phi(t+1) = \phi(t) - \alpha \nabla_{\phi(t)} L(\phi)(t) \tag{6.15}$$

6.3.2 变分 E-M

E-M 算法和变分推断可以统一到 “变分 E-M” 的框架中。它们都希望通过优化一个形式上比对数似然更好的函数来间接达到优化似然，或使用另一族函数逼近似然的目的。

具体来说，假设有参数 θ 控制的含隐变量的概率分布 $p(\mathbf{x}|\mathbf{z}, \theta)$。对隐变量后验分布 $p(\mathbf{z}|\mathbf{x}, \theta)$，有参数 ϕ 控制的近似分布 $q(\mathbf{z}|\mathbf{x}, \phi)$。那么，我们有恒等式

$$\begin{aligned}
\log L(\theta) &= \left\langle \log \frac{p(\mathbf{x}, \mathbf{z}|\theta)}{q(\mathbf{z}|\mathbf{x}, \phi)} \right\rangle_{q(\mathbf{z}|\mathbf{x}, \phi)} + \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \theta)] \\
&= \langle \log p(\mathbf{x}, \mathbf{z}|\theta) \rangle_{q(\mathbf{z}|\mathbf{x}, \phi)} + \mathrm{H}[q(\mathbf{z}|\mathbf{x}, \phi)] + \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \theta)] \\
&= \langle \log p(\mathbf{x}|\mathbf{z}, \theta) \rangle_{q(\mathbf{z}|\mathbf{x}, \phi)} - \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\theta)] + \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \theta)]
\end{aligned} \tag{6.16}$$

首先，由于 KL 散度的非负性，我们可以得到经典的 ELBO，即证据下界

$$\log L(\theta) \geq \mathrm{ELBO}(\theta) = \left\langle \log \frac{p(\mathbf{x}, \mathbf{z}|\theta)}{q(\mathbf{z}|\mathbf{x}, \phi)} \right\rangle_{q(\mathbf{z}|\mathbf{x},\phi)}$$

在 M 步取 ELBO 为优化目标，就可以得到变分推断的出发条件

$$\max_{\theta} \mathrm{ELBO}(\theta) = \left\langle \log \frac{p(\mathbf{x}, \mathbf{z}|\theta)}{q(\mathbf{z}|\mathbf{x}, \phi)} \right\rangle_{q(\mathbf{z}|\mathbf{x},\phi)} \tag{6.17}$$

其次，由于分布的信息熵也非负，而且 $q(\mathbf{z}|\mathbf{x}, \phi)$ 与 θ 无关，因此有

$$\log L(\theta) \geq LB(\theta) = \langle \log p(\mathbf{x}, \mathbf{z}|\theta) \rangle_{q(\mathbf{z}|\mathbf{x},\phi)}$$

取 $LB(\theta)$ 为优化目标，并令 $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{x}, \mathbf{z}|\theta_{old})$，就得到经典的 E-M 算法

$$\max_{\theta} LB(\theta) = \langle \log p(\mathbf{x}, \mathbf{z}|\theta) \rangle_{p(\mathbf{x},\mathbf{z}|\theta_{old})} \tag{6.18}$$

虽然在式6.16中，$-\mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\theta)]$ 一项是负的，而且也和 θ 有关，但这一项中 $p(\mathbf{z}|\theta)$ 的 θ 可取上一轮迭代的 θ_{old}，可以视为常数。这样，优化条件概率对数也可以起到优化似然的作用。只不过，它并不是真正的下界，我们称之为条件函数 CF

$$\max_{\theta} CF(\theta) = \langle \log p(\mathbf{x}|\mathbf{z}, \theta) \rangle_{q(\mathbf{z}|\mathbf{x},\phi)}$$

令 $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{x}, \mathbf{z}|\theta_{old})$，得到以条件函数为优化目标的 EM 算法

$$\max_{\theta} CF(\theta) = \langle \log p(\mathbf{x}|\mathbf{z}, \theta) \rangle_{p(\mathbf{x},\mathbf{z}|\theta_{old})} \tag{6.19}$$

对于下界 $LB(\theta)$，注意到

$$p(\mathbf{z}|\mathbf{x}, \theta_{old}) = \frac{p(\mathbf{x}, \mathbf{z}|\theta_{old})}{p(\mathbf{x}|\theta_{old})}$$

其分母与隐变量无关。因此可以将分母从期望式中取出，得到

$$\log L(\theta) \geq p(\mathbf{x}|\theta_{old}) \langle \log p(\mathbf{x}, \mathbf{z}|\theta) \rangle_{p(\mathbf{x},\mathbf{z}|\theta_{old})}$$

这样，在 M 步我们可以只优化和 $\mathbf{z}$ 的部分。称这部分为熵函数 HF。它不是似然函数的下界，但可以作为 M 步的优化对象，即

$$\max_{\theta} HF(\theta) = \langle \log p(\mathbf{x}, \mathbf{z}|\theta) \rangle_{p(\mathbf{x},\mathbf{z}|\theta_{old})} \tag{6.20}$$

注意上式不是某个分布真正的熵，只是具有熵的形式。

上述多样的优化形式可总结为表6.3.2

目标函数	表达式	是否为下界	算法类型	例子
$\mathrm{ELBO}(\theta)$	$\left\langle \log \frac{p(\mathbf{x}, \mathbf{z}\|\theta)}{q(\mathbf{z}\|\mathbf{x}, \phi)} \right\rangle_{q(\mathbf{z}\|\mathbf{x},\phi)}$	是	VI	BM(10.2.2), VAE(12.1)
$LB(\theta)$	$\langle \log p(\mathbf{x}, \mathbf{z}\|\theta) \rangle_{p(\mathbf{z}\|\mathbf{x},\theta_{old})}$	是	EM	线性回归 (7.2.6), GMM(8.3.1)
$CF(\theta)$	$\langle \log p(\mathbf{x}\|\mathbf{z}, \theta) \rangle_{p(\mathbf{z}\|\mathbf{x},\theta_{old})}$	否	EM	因子分析 (8.4.2)
$HF(\theta)$	$\langle \log p(\mathbf{x}, \mathbf{z}\|\theta) \rangle_{p(\mathbf{x},\mathbf{z}\|\theta_{old})}$	否	EM	HMM(11.2.5)

6.3.3 局部变分近似

凸函数局部下界

在变分近似中，需要对一个函数求局部下界。即：

$$\text{对于函数 } f(x)\text{，求函数 } g(x) \text{ 使得 } f(x) \geq g(x), \forall x$$

一种通常的方法是，对具有凸性的函数，用一条直线做近似。例如，假设函数 $f(\mathbf{x})$ 满足 $\nabla^2 f \geq 0$，则对于一定范围内的 λ，函数 $\lambda x + b - f(x)$ 对 x 有最大值。即：

$$\lambda x + b - f(x) \leq \max_y \{\lambda y + b - f(y)\}$$

其中 b 是和 x 无关的项。消去 b，进行移项，有

$$f(x) \geq \lambda x - \max_y \{\lambda y - f(y)\}$$

也就是说，取 $b = -\max_y \{\lambda y - f(y)\}$，则我们找到了满足条件的 $g(x)$

$$g(x) = \lambda x - \max_y \{\lambda y - f(y)\} \tag{6.21}$$

sigmoid 函数下界

下面我们考虑对不满足凸性的函数寻找下界。以 sigmoid 函数 $\text{sig}(x) = (1 + \exp(-x))^{-1}$ 为例。对其取对数，设

$$f(x) = \log \text{sig}(x) - \frac{x}{2} = -\log(\exp(\frac{x}{2}) + \exp(-\frac{x}{2}))$$

$f(x)$ 仍然没有凸性。再令 $z(x) = x^2$，设

$$h(z) = f(x(z)) = f(\sqrt{z}) = -\log(\exp(\frac{\sqrt{z}}{2}) + \exp(-\frac{\sqrt{z}}{2}))$$

可以证明对 $z \geq 0$ 有 $\nabla^2 h(z) \geq 0$。

这样，如果我们先找到 $h(z)$ 的下界 $k(z)$，就能得到 $f(x)$ 的下界 $l(x)$，从而得到 $\text{sig}(x)$ 的下界 $g(x)$。推导如下：

由于

$$h(z) \geq k(z),\ \forall z \geq 0$$

取 $l(x) = k(x^2)$，就有

$$f(x) = h(z(x)) = h(x^2) \geq k(x^2) = k(z(x)) = l(x),\ \forall x$$

因此

$$\text{sig}(x) = \exp f(x) + \frac{x}{2} \geq \exp l(x) + \frac{x}{2} = \exp k(x^2) + \frac{x}{2},\ \forall x$$

即

$$\text{sig}(x) \geq g(x) = \exp\{k(x^2) + \frac{x}{2}\},\ \forall x$$

下面来求 $h(z)=-\log(\exp(\frac{\sqrt{z}}{2})+\exp(-\frac{\sqrt{z}}{2}))$ 的下界 $k(z)$。根据式6.21，有

$$k(z)=\eta z-\max_{y}\{\eta y-h(y)\}$$

设函数 $m(y)=\eta y-h(y)$，其取到最大值的 y 记作 y_{max}。根据凸函数性质，有

$$m'(y_{max})=\eta+\frac{1}{4\sqrt{y_{max}}}\frac{\exp(\sqrt{y_{max}})-1}{\exp(\sqrt{y_{max}})+1}=0$$

记 $\xi=\sqrt{y_{max}}$，则有

$$\eta=-\frac{1}{4\xi}\frac{\exp(\xi)-1}{\exp(\xi)+1}=-\frac{1}{4\xi}(2\,\mathrm{sig}(\xi)-1)$$

因此可以写出 $k(z)$ 的表达式

$$k(z)=\eta z-\eta y_{max}+h(y_{max})=\eta z-\eta y_{max}-\log\left(\exp\left\{\frac{\sqrt{y_{max}}}{2}\right\}+\exp\left\{-\frac{\sqrt{y_{max}}}{2}\right\}\right)$$

为表达方便可以写成 ξ 的形式

$$k(z)=\eta z-\eta\xi^2-\log\left(\exp\left\{\frac{\xi}{2}\right\}+\exp\left\{-\frac{\xi}{2}\right\}\right)$$

因此 $\mathrm{sig}(x)$ 的下界为

$$\begin{aligned}\mathrm{sig}(x)\geq& g(x)=\exp(k(x^2)+\frac{x}{2})\\&=\exp\{\eta x^2-\eta\xi^2-\log(\exp(\frac{\xi}{2})+\exp(-\frac{\xi}{2}))+\frac{x}{2}\}\\&=\mathrm{sig}(\xi)\exp\{\frac{x-\xi}{2}+\eta(x^2-\xi^2)\}\end{aligned}$$

其中 η 和 ξ 满足关系

$$\eta(\xi)=-\frac{1}{4\xi}(2\,\mathrm{sig}(\xi)-1)$$

也有文献取 $\lambda=-\eta$，即

$$\begin{aligned}\mathrm{sig}(x)\geq&\,\mathrm{sig}(\xi)\exp\left(\frac{x-\xi}{2}-\lambda(\xi)(x^2-\xi^2)\right),\ \forall x\\\lambda(\xi)=&\frac{1}{4\xi}(2\,\mathrm{sig}(\xi)-1)\end{aligned}\tag{6.22}$$

在式 6.22中，ξ 可以取 R 中的任何实数，得到不同的对 $\mathrm{sig}(x)$ 的高斯形式的下界估计。估计函数在 $x=\pm\xi$ 处可以取等号。

6.3.4 拉普拉斯近似

算法	**拉普拉斯近似**
算法简述	对于参数分布求解问题，使用高斯分布进行近似
已知	最大似然参数 $\mathbf{w}_{MAP}$，后验分布 $p(\mathbf{w}\|D)$ 的形式
求	近似高斯分布参数 μ_L, Σ_L
解类型	近似解
近似解	$\Sigma_L = -(\nabla^2 f_{MAP})^{-1}$ $\mu_L = \mathbf{w}_{MAP}$

概率方法中经常需要求解某些条件分布或后验分布 $p(\mathbf{w}|D)$，或者需要对它进行积分。有时候，这一分布的精确形式过于复杂，此时可以考虑用一个相同维度的高斯分布进行近似。

$$p(\mathbf{w}|D) \approx \tilde{p}(\mathbf{w}) = N(\mathbf{w}; \mu_L, \Sigma_L)$$

假设原概率分布是指数族乘积形式，对等号两边取对数有

$$f(\mathbf{w}) = \log p(\mathbf{w}|D) \approx \tilde{\log} p(\mathbf{w}) = -\frac{1}{2}(\mathbf{w}-\mu_L)^T \Sigma_L^{-1}(\mathbf{w}-\mu_L) + C$$

其中 C 是和 $\mathbf{w}$ 无关的常数。我们希望近似分布 $\tilde{p}(\mathbf{w})$ 也在最大后验解 $\mathbf{w}_{MAP}$ 处取到最大值，即

$$\mu_L = \mathbf{w}_{MAP}$$

因此，我们要对 $f(\mathbf{w})$ 求二阶导，在最大似然解附近进行泰勒展开。此时一阶导数为 $\mathbf{0}$，记 $f(\mathbf{w})$ 在 $\mathbf{w}_{MAP}$ 处的二阶导为 $\nabla^2 f_{MAP}$，则有

$$f(\mathbf{w}) \approx f(\mathbf{w}_{MAP}) + \frac{1}{2}(\mathbf{w}-\mathbf{w}_{MAP})^T \nabla^2 f_{MAP}(\mathbf{w}-\mathbf{w}_{MAP})$$

即得到拉普拉斯近似

$$\begin{aligned} \Sigma_L &= -(\nabla^2 f_{MAP})^{-1} \\ \mu_L &= \mathbf{w}_{MAP} \end{aligned} \tag{6.23}$$

6.4 采样方法

在很多无法进行精确积分的场景下，尤其是和概率密度相关的积分场景下，采样方法是相当重要的近似方法。和前面介绍的近似方法不同，采样方法属于随机性近似。

可以如下形式化地定义采样近似。设有随机变量服从概率分布 $\mathbf{x} \sim p(\mathbf{x})$，对于函数 $f(\mathbf{x})$，期望 $E_{x\sim p}[f(\mathbf{x})]$ 的一种近似是

$$E_{x\sim p}[f(\mathbf{x})] \approx \sum_{i=1}^{N} f(\mathbf{x}^{(i)})$$

其中 $\mathbf{x}^{(i)} \sim p(\mathbf{x})$ 是一组独立同分布 p 的随机变量，称为一组 p 的采样。本节关心的问题就是：如何获取满足要求的采样。

6.4.1 直接采样

由于离散分布的采样较为简单，只需对均匀分布按概率质量划分即可。在此只讨论连续分布。

算法	**直接采样**
算法简述	对于采样问题，使用均匀分布采样通过分布函数逆函数变换得到样本
已知	一维连续分布 $p(x)$，分布函数 $F(x)$ 可逆
求	采样样本 $x^{(1)}, x^{(2)}, ..., x^{(N)} \sim p(x)$
采样方法	从均匀分布 $U[0,1]$ 中采样得到 $y^{(1)}, y^{(2)}, ..., y^{(N)}$，$x^{(i)} = F^{-1}(y^{(i)})$

考虑一维连续已知分布 $p(x)$，可以求出其分布函数

$$F(x) = \int_{-\infty}^{x} p(z)dz$$

该分布函数是一个以 $(-\infty, +\infty)$ 为定义域，$[0,1]$ 为值域的一元严格单调增函数，且满足

$$\lim_{x \to -\infty} F(x) = 0$$

$$\lim_{x \to +\infty} F(x) = 1$$

则该函数存在反函数 $F^{-1}(y) = G(y) = x$。

假设可以方便地从 $[0,1]$ 上的均匀分布中采样 $\mathbf{y}(1), y^{(2)}, .., y^{(N)}$，即

$$p(y^{(i)} \leq \mathbf{y}) = y,\ y \in [0,1]$$

由单调性，则有

$$p(G(y^{(i)}) \leq G(y)) = p(x^{(i)} < y) = y,\ y \in [0,1]$$

设 $x = G(y)$，即 $y = F(x)$，则有

$$p(G(y^{(i)}) \leq x) = F(x),\ \forall x$$

因此 $G(y^{(1)}), G(y^{(2)}), ..., G(y^{(N)})$ 是服从分布 $p(x)$ 的独立同分布序列。

对于多维分布，如 $p(\mathbf{x}) = N(\mathbf{x}; \mu, \Sigma)$，可以首先对同维度的标准正态分布进行采样，再通过线性变换得到目标分布的样本。由于标准正态分布每维度均独立，因此可以对每个维度分别使用一维采样方法。

6.4.2 接受-拒绝采样

简单采样方法需要满足两个苛刻的条件，即密度函数可积以及分布函数严格单增，来保证反函数一定存在。如果不满足这样的条件，就需要其他的采样方法。

算法	**接受-拒绝采样**
算法简述	对于采样问题，先从提议分布采集样本， 再以提议概率和目标概率的比值为概率接受样本
已知	分布 $p(\mathbf{x})$，简单分布 $q(\mathbf{x})$
求	采样样本 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)} \sim p(\mathbf{x})$
采样方法	1. 计算 $c = \min_{\mathbf{x}} \frac{p(\mathbf{x})}{q(\mathbf{x})}$ 2. 从 $q(\mathbf{x})$ 中采到样本 $\mathbf{y}^{(i)}$ 3. 从均匀分布 $U(0,1)$ 中采到样本 $u^{(i)}$ 4. 若 $u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}$ 则将 $\mathbf{y}^{(i)}$ 加入样本集，否则丢弃样本 5. 若样本集数量为 N，停止采样，否则回第 2 步

具体来说，对于难以直接采样的分布 $p(\mathbf{x})$，假设存在简单的、容易采样的分布 $q(\mathbf{x})$ 以及常数 $c > 0$，满足

$$c = \min_{x} \frac{p(\mathbf{x})}{q(\mathbf{x})}$$

此时，可以从 $q(\mathbf{x})$ 中采样得到 $\mathbf{y}^{(i)}$，并计算 $\frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}$。接下来，从 $U[0,1]$ 中采样得到 $u^{(i)}$。若

$$u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}$$

则将 $\mathbf{y}^{(i)}$ 加入样本集，称为“接受”。否则，丢弃样本继续进行采样，称作“拒绝”。这里的分布 $q(\mathbf{x})$ 称为提议分布。

该算法的正确性证明如下。假设 $p(\mathbf{x}), q(\mathbf{x})$ 对应分布函数为 $F(\mathbf{x}), G(\mathbf{x})$，考虑如下条件概率

$$p\left(\mathbf{y}^{(i)} \leq \mathbf{y} | u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}\right) = \frac{p\left(\mathbf{y}^{(i)} \leq \mathbf{y}, u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}\right)}{p\left(u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}\right)}$$

其中

$$\begin{aligned}
& p\left(u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}\right) \\
=& \int p\left(u^{(i)} \leq \frac{p(\mathbf{y})}{cq(\mathbf{y})} | \mathbf{y}^{(i)} = \mathbf{y}\right) p(\mathbf{y}^{(i)} = \mathbf{y}) d\mathbf{y} \\
=& \int p\left(u^{(i)} \leq \frac{p(\mathbf{y})}{cq(\mathbf{y})}\right) q(\mathbf{y}) d\mathbf{y} \\
=& \int \frac{p(\mathbf{y})}{cq(\mathbf{y})} q(\mathbf{y}) d\mathbf{y} \\
=& \frac{1}{c} \int p(\mathbf{y}) d\mathbf{y} = \frac{1}{c}
\end{aligned}$$

因此

$$
\begin{aligned}
& p\left(\mathbf{y}^{(i)} \leq \mathbf{y} | u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}\right) \\
&= \frac{p\left(\mathbf{y}^{(i)} \leq \mathbf{y}, u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}\right)}{p\left(u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}\right)} \\
&= c\, p\left(u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})}, \mathbf{y}^{(i)} \leq \mathbf{y}\right) \\
&= c \int_{-\infty}^{y} p\left(u^{(i)} \leq \frac{p(\mathbf{y}^{(i)})}{cq(\mathbf{y}^{(i)})} | \mathbf{y}^{(i)} = \mathbf{w}\right) p(y^{(i)} = \mathbf{w}) d\mathbf{w} \\
&= c \int_{-\infty}^{y} p\left(u^{(i)} \leq \frac{p(\mathbf{w})}{cq(\mathbf{w})}\right) q(\mathbf{w}) d\mathbf{w} \\
&= c \int_{-\infty}^{y} \frac{p(\mathbf{w})}{cq(\mathbf{w})} q(\mathbf{w}) d\mathbf{w} \\
&= \int_{-\infty}^{y} p(\mathbf{w}) d\mathbf{w} = F(y)
\end{aligned}
$$

接受-拒绝采样可以应用于高维空间的分布，但容易出现拒绝率高的问题。原因是高维空间分布 $q(\mathbf{x})$ 的质量一般集中于空间中某一体积较小的区域，该区域若不被 $p(\mathbf{x})$ 对应的可行区域涵盖，拒绝概率就会很大。因此，对于高维数据，还需要更加方便的采样方法。

6.4.3 MCMC 采样

算法	**MCMC 采样**
算法简述	对于采样问题，通过构造以目标分布为平稳分布的马尔科夫链，从稳定步骤采集样本
已知	概率分布 $\pi(\mathbf{x})$，转移分布 $q(\mathbf{a}, \mathbf{a}')$
求	采样样本 $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)} | \mathbf{x}^{(i)} \sim \pi(\mathbf{x})\}$
采样方法	1. 取形式简单的转移分布 $p(\mathbf{x}(t+1) = \mathbf{a} | \mathbf{x}(t) = \mathbf{a}') = q(\mathbf{a}, \mathbf{a}')$， 　　计算接受函数 $\alpha(\mathbf{a}, \mathbf{a}') = q(\mathbf{a}', \mathbf{a})\pi(\mathbf{a}')$ 2. 任取初始状态 $\mathbf{x}_0$, 进行充分多次转移采样构成马尔可夫链 $\mathbf{x}(t+1) \sim q(\mathbf{x}(\mathbf{t}+\mathbf{1}), \mathbf{x}(\mathbf{t}))$ 3. 从均匀分布中采样 $u \sim U[0,1]$，从转移分布采样 $\mathbf{y} \sim q(\mathbf{x}(\mathbf{t}+\mathbf{1}), \mathbf{x}(\mathbf{t}))$ 4. 若 $u \leq \alpha(\mathbf{y}, \mathbf{x}(\mathbf{t}))$，则 $\mathbf{x}(\mathbf{t}+\mathbf{1}) = \mathbf{y}$，且加入样本集 $\mathbf{X}$；否则，回到第 3 步 5. 若样本集数量为 N，停止采样；否则回到第 3 步

对于一个概率转移矩阵 P 已知的马尔科夫链，若所有状态满足非周期性和连通性，则一定存在状态平稳分布 $\pi(\mathbf{x})$，即

$$\pi P = \pi$$

平稳分布满足一个性质：对于服从任意分布的初始状态 X_0，存在有限整数 N，可以认为转移序列 $\{X_N\}$ 从第 N 步开始的随机变量 $X_N, X_{N+1}, X_{N+2}...$ 均服从平稳分布 $\pi(\mathbf{x})$。

因此，对任意概率分布 $\pi(\mathbf{x})$ 而言，如果能找到以 $\pi(\mathbf{x})$ 为平稳分布的状态转移矩阵 $P(\pi)$，就可以使用转移序列实现 $\pi(\mathbf{x})$ 的抽样。

对于马尔科夫链平稳分布 $\pi(\mathbf{x})$ 和状态转移矩阵 P，有如下的细致平稳条件：

$$\pi(i)P(i,j) = \pi(j)P(j,i), \forall i,j \tag{6.24}$$

对于多维状态的马尔可夫链，细致平稳条件是 $\pi(\mathbf{x})$ 是平稳分布的一个充分条件，因为

$$\pi(j) = \sum_i \pi(j)P(j,i) = \sum_i \pi(i)P(i,j) = (\pi\mathbf{P})_j$$

对于一般概率转移矩阵 Q，可以使用接受-拒绝采样的思想使实际采样的概率转移矩阵满足细致平稳条件。对于任意状态 i,j，取接受率

$$\alpha(i,j) = Q(j,i)\pi(j) \tag{6.25}$$

则 $\alpha(i,j) \in [0,1]$，且满足

$$\pi(i)Q(i,j)\alpha(i,j) = \pi(i)Q(i,j)Q(j,i)\pi(j) = \pi(j)Q(j,i)\alpha(j,i) \tag{6.26}$$

对于马尔科夫链上任一当前状态 $X(t)$，从均匀分布中随机抽样 $U \sim U[0,1]$。若 $u^{(i)} \leq \alpha(i,j)$，则按照 Q 矩阵进行转移。否则，再次尝试抽样。这样有

$$p(X(t+1) = j | X(t) = i, u^{(i)} \leq \alpha(i,j)) = Q(i,j)$$

就有

$$\begin{aligned}\hat{P}(i,j) &= p\big(X(t+1) = j | X(t) = i, u^{(i)} \leq \alpha(i,j)\big) \\ &= p\big(X(t+1) = j | X(t) = i\big) p\big(u^{(i)} \leq \alpha(i,j)\big) = Q(i,j)\alpha(i,j)\end{aligned}$$

因此

$$\pi(i)\hat{P}(i,j) = \pi(i)Q(i,j)\alpha(i,j) = \pi(j)Q(j,i)\alpha(j,i) = \pi(j)\hat{P}(j,i)$$

满足细致平稳条件。根据上一小节的证明，这样抽出的样本服从平稳分布 $p(\mathbf{x})$。

以上推导中，假定了随机变量 X 的取值是离散有限的。对于连续随机变量 $\mathbf{x}$，需要将概率转移矩阵 $P(i,j)$ 理解为转移概率分布 $p(X(t+1)|X(t))$。

6.4.4 M-H 采样

算法	**M-H 采样**
算法简述	对于采样问题，通过构造以目标分布为平稳分布的马尔科夫链，从稳定步骤采集样本
已知	概率分布 $\pi(\mathbf{x})$，转移分布 $q(\mathbf{a},\mathbf{a}')$
求	采样样本 $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)} | \mathbf{x}^{(i)} \sim \pi(\mathbf{x})\}$
采样方法	1. 取形式简单的转移分布 $p(\mathbf{x}(t+1) = \mathbf{a} | \mathbf{x}(t) = \mathbf{a}') = q(\mathbf{a},\mathbf{a}')$， 计算接受函数 $\alpha(\mathbf{a},\mathbf{a}') = \min\left\{1, \frac{q(\mathbf{a}',\mathbf{a})\pi(\mathbf{a}')}{q(\mathbf{a},\mathbf{a}')\pi(\mathbf{a})}\right\}$ 2. 任取初始状态 $\mathbf{x}_0$，进行充分多次转移采样构成马尔可夫链 $\mathbf{x}(t+1) \sim q(\mathbf{x}(\mathbf{t}+\mathbf{1}), \mathbf{x}(\mathbf{t}))$ 3. 从均匀分布中采样 $u \sim U[0,1]$，从转移分布采样 $\mathbf{y} \sim q(\mathbf{x}(\mathbf{t}+\mathbf{1}), \mathbf{x}(\mathbf{t}))$ 4. 若 $u \leq \alpha(\mathbf{y}, \mathbf{x}(\mathbf{t}))$，则 $\mathbf{x}(\mathbf{t}+\mathbf{1}) = \mathbf{y}$，且加入样本集 $\mathbf{X}$；否则，回到第 3 步 5. 若样本集数量为 N，停止采样；否则回到第 3 步

尽管 MCMC 采样能满足细致平稳条件，但仍然存在接受率低的问题。原因是接受率 $\alpha(i,j)=Q(j,i)\pi(j)$ 是两项概率乘积，而这两项可能各自都很小。

因此，解决的方法是：在保证 6.26的条件下，增大 α 的绝对数值。由于 α 不能超过 1，因此我们将 $Q(j,i)\pi(j)$ 和 $Q(i,j)\pi(i)$ 同时扩倍，令更大的那个扩倍为 1，即

$$\alpha(i,j)=\begin{cases}1 & \text{if } Q(j,i)\pi(j)\geq Q(i,j)\pi(i)\\ \dfrac{Q(j,i)\pi(j)}{Q(i,j)\pi(i)} & \text{if } Q(j,i)\pi(j)<Q(i,j)\pi(i)\end{cases}$$

或

$$\alpha(i,j)=\min\left\{1,\frac{Q(j,i)\pi(j)}{Q(i,j)\pi(i)}\right\} \tag{6.27}$$

这种采样方法被称为 Metropolis-Hastings 采样，简称 M-H 采样。

6.4.5 Gibbs 采样

算法	**Gibbs 采样**
算法简述	对于多维分布采样问题，依次从各维度按条件概率采样
已知	概率分布 $\pi(\mathbf{x})$，保证收敛的迭代轮次 M
求	采样样本 $\mathbf{X}=\{\mathbf{x}^{(1)},\mathbf{x}^{(2)},...,\mathbf{x}^{(N)}\|\mathbf{x}^{(i)}\sim\pi(\mathbf{x})\}$
采样方法	1. 随机初始化初始状态 $\mathbf{x}(0,0)$ 2. 对于当前状态 $\mathbf{x}(t,d)$，若 d 是最后一维，进入第 3 步. 否则，根据第 $d+1$ 维的条件分布采样下一状态的 $d+1$ 维 $x_{d+1}(t,d+1)\sim\pi_{d+1\|-(d+1)}(x_{d+1}\|\mathbf{x}_{-d+1})$， 保持其余维度不变 $\mathbf{x}_{-(d+1)}(t,d+1)=\mathbf{x}_{-(d+1)}(t,d)$ 3. 不论是否有 $t>M$，设置 $\mathbf{x}(t+1,0)=\mathbf{x}(t,d)$。 若 $t>M$，将 $\mathbf{x}(t,d)$ 加入样本集 $\mathbf{X}$。 4. 若样本集数量足够，停止采样。否则，回到第 2 步。

对于高维变量 $\mathbf{x}$，有时直接对联合分布采样较为复杂，但对各维度的条件分布，即一维分布采样简单。如果能将联合分布采样分解为每个维度的采样，将会有效简化计算过程。

对于多维随机变量 $\mathbf{x}$，记它的第 i 个维度为 x_i，除了第 i 维的其他维度为 $\mathbf{x}_{-i}$。由联合分布，我们可以计算出第 i 个维度的条件分布

$$\pi_{i|-i}(x_i|\mathbf{x}_{-i})=\frac{\pi(x_i,\mathbf{x}_{-i})}{\pi_{-i}(\mathbf{x}_{-i})}=\frac{\pi(x_i,\mathbf{x}_{-i})}{\int\pi(x_i,\mathbf{x}_{-i})dx_i}$$

对于多维随机变量，考虑固定除第 i 个维度以外的其他维度，仅改变第 i 个维度的条件转移，即 $\mathbf{a'}_{-i}=\mathbf{a}_{-i}$，且转移分布

$$p(\mathbf{a},\mathbf{a}')=\pi_{i|-i}(a'_i|\mathbf{a}_{-i}) \tag{6.28}$$

这样的分布天然满足细致平稳条件

$$
\begin{aligned}
&\pi(\mathbf{a})p(\mathbf{a},\mathbf{a}') \\
=&\pi(a_i,\mathbf{a}_{-i})\pi_{i|-i}({a'}_i|\mathbf{a}_{-i}) \\
=&\frac{\pi(a_i,\mathbf{a}'_{-i})\pi({a'}_i,\mathbf{a}_{-i})}{\pi_{-i}(\mathbf{a}_{-i})} \\
=&\frac{\pi(a_i,\mathbf{a}'_{-i})\pi({a'}_i,\mathbf{a}_{-i})}{\pi_{-i}(\mathbf{a}'_{-i})} \\
=&\pi_{i|-i}(a_i|\mathbf{a}'_{-i})\pi({a'}_i,\mathbf{a}'_{-i}) \\
=&\pi(\mathbf{a}')p(\mathbf{a}',\mathbf{a})
\end{aligned}
$$

因此，利用这种特点，轮流按条件分布从各个维度采样，就可以得到平稳分布为 $\pi(\mathbf{x})$ 的马尔可夫过程，从而完成从 $\pi(\mathbf{x})$ 中采样。

6.4.6 重要性采样

算法	**重要性采样**
算法简述	对于采样求积分问题，根据实际分布和参考分布比值对参考分布样本加权
已知	概率分布 $\pi(\mathbf{x})$, 参考分布 $q(\mathbf{x})$
求	积分 $\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$ 的采样近似
求解方法	$\int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \approx \sum_{i=1}^{N} f(\mathbf{x}^{(i)})\frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})},\ \mathbf{x}^{(i)} \sim q(\mathbf{x})$

很多时候，对分布采样的目的不是为了获得一组样本，而是近似计算某个积分

$$
\int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} = E_{x\sim\pi}[f(\mathbf{x})] \approx \sum_{i=1}^{N} f(\mathbf{x}^{(i)})
$$

此时如果目标分布不方便采样，可以从参考分布采样并加权

$$
\begin{aligned}
&\int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \\
=&\int f(\mathbf{x})\frac{\pi(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x} \\
=&E_{x\sim q}[f(\mathbf{x})\frac{\pi(\mathbf{x})}{q(\mathbf{x})}] \\
\approx&\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})},\ \mathbf{x}^{(i)} \sim q(\mathbf{x})
\end{aligned}
$$

即

$$
\int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \approx \sum_{i=1}^{N} f(\mathbf{x}^{(i)})\frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})},\ \mathbf{x}^{(i)} \sim q(\mathbf{x}) \tag{6.29}
$$

7 基础概率模型

本章中将介绍一些较为基础但十分有效的概率机器学习模型，用于解决分类和回归问题。其中的一些模型可能已经在第3章介绍过，本章中将提供一些新的视角和理解方式。

7.1 朴素贝叶斯

分类问题中有一类特殊的情形：样本 $\mathbf{x}$ 的所有分量和标签 y 均为离散变量，即它们的取值范围都是有限集合。此时，朴素贝叶斯模型十分有效。

算法	**朴素贝叶斯**
算法简述	对于分类问题，若样本和标签全为有限离散值， 假设样本各维度互相独立，求判别模型
已知	样本 $\{\mathbf{x}^{(i)}\|x_j^{(i)} \in \mathbf{C}_j, i = 1, 2, ..., N\}$，标签 $\{y^{(i)} \in \mathbf{Y}\|i = 1, 2, ..., N\}$
求	后验概率 $p(y\|\mathbf{x})$
解类型	闭式解
闭式解	$p(y\|\mathbf{x}) = \frac{\#(Y=y)}{N}\prod_{j=1}^{n}\frac{\#(X_j=x_j,Y=y)}{\#(Y=y)}\prod_{k=1}^{n}\frac{N}{\#(X_k=x_k)}$

对于分类问题，我们假设样本各维度互相独立。由贝叶斯公式，后验概率可以分解为

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} = \frac{p(y)\prod_{j=1}^{n} p(\mathbf{x}_j|y)}{p(\mathbf{x})}$$

其中 X_j 表示样本第 j 维对应随机变量，Y 表示 y 对应随机变量。记示性函数为 $I(\cdot)$。根据频率学派观点，概率可以用频率近似，则有

$$\begin{aligned}
&p(y|\mathbf{x}) = \frac{p(y)\prod_{j=1}^{n} p(\mathbf{x}_j|y)}{p(\mathbf{x})} \\
=&\frac{\sum_{i=1}^{N} I(y^{(i)}=y)}{N}\prod_{j=1}^{n}\frac{\sum_{i=1}^{N} I(x_j^{(i)}=x_j, y^{(i)}=y)}{\sum_{i=1}^{N} I(y^{(i)}=y)}\prod_{k=1}^{n}\frac{N}{\sum_{i=1}^{N} I(x_k^{(i)}=x_k)} \\
=&\frac{\#(Y=y)}{N}\prod_{j=1}^{n}\frac{\#(X_j=x_j,Y=y)}{\#(Y=y)}\prod_{k=1}^{n}\frac{N}{\#(X_k=x_k)}
\end{aligned} \tag{7.1}$$

其中 $\#(\cdot)$ 表示在数据集中对满足括号内条件的样本计数。

7.2 概率线性回归

线性回归已经在第 3 章中介绍过，但概率方法有着不同的视角。

设有样本集 $X \in R^{n\times N}$，标签集 $Y \in R^N$。假设标签作为连续随机变量服从一个高斯分布，但其均值被样本 $\mathbf{x}$ 和参数 $\mathbf{w} \in R^n$ 以线性方式控制，即

$$p(y|\mathbf{x}, \mathbf{w}) = N(y; \mathbf{w}^T\mathbf{x}, \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}}\exp\{-\beta(y - \mathbf{w}^T\mathbf{x})^2\}$$

其中 β 是一个超参数，含义为精度（即方差的倒数）。那么问题就变成，如何根据数据集 $D = \{X, Y\}$ 求 $\mathbf{w}$。

7.2.1 最大似然估计

算法	**线性回归-最大似然**
算法简述	对于回归问题，假设标签服从高斯分布，其均值为样本的线性变换，求出最大似然系数
已知	样本 $X \in R^{n \times N}$（扩充过），标签 $Y \in R^N$
求	线性系数 $\mathbf{w} \in R^n$ 使得 $\max_{\mathbf{w}} L(w\|D) = \prod_{i=1}^{N} p(y^{(i)}\|\mathbf{x}^{(i)}, \mathbf{w})$
解类型	闭式解
闭式解	$\mathbf{w}^\star = (X^T X)^{-1} X^T \mathbf{y}$

一种合理的想法是，最大化参数在数据集上的似然函数，即

$$\max_{\mathbf{w}} L(w|D) = \prod_{i=1}^{N} p(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w})$$

最大化似然相当于最小化负对数似然，即

$$\begin{aligned}
&\max_{\mathbf{w}} -\log L(w|D) \\
=&\max_{\mathbf{w}} -\log \prod_{i=1}^{N} p(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w}) \\
=&\max_{\mathbf{w}} -\sum_{i=1}^{N} \log \sqrt{\frac{\beta}{2\pi}} \exp\left\{-\frac{\beta}{2}(y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2\right\} \\
=&\max_{\mathbf{w}} \frac{\beta}{2} \sum_{i=1}^{N} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2
\end{aligned}$$

可以看到，最大似然和使用平方误差给出的损失函数完全一致。这是一个最小二乘问题，可以直接求出闭式解。

$$\mathbf{w}^\star = (X^T X)^{-1} X^T \mathbf{y} \tag{7.2}$$

7.2.2 最大后验估计

算法	**线性回归-最大后验**
算法简述	对于回归问题，假设标签服从高斯分布，其均值为样本的线性变换， 参数服从高斯先验，求出最大后验系数
已知	样本 $X \in R^{n \times N}$（扩充过），标签 $Y \in R^N$，先验 $p(\mathbf{w}\|\alpha) = N(\mathbf{w}; \mathbf{0}, \alpha^{-1} I)$
求	线性系数 $\mathbf{w} \in R^n$ 使得 $\max_{\mathbf{w}} p(\mathbf{w}\|D) = \frac{1}{Z} p(\mathbf{w}\|\alpha) \prod_{i=1}^{N} p(y^{(i)}\|\mathbf{x}^{(i)}, \mathbf{w})$
解类型	闭式解
闭式解	$\mathbf{w}^\star = (X^T X + \alpha I)^{-1} X^T \mathbf{y}$

最大似然没有考虑参数的先验分布问题。在上面的高斯模型中，似然函数是高斯的形式，其关于均值的共轭分布也是高斯分布。因此，可以假设参数 $\mathbf{w}$ 服从 0 均值的高斯分布

$$p(\mathbf{w}|\alpha) = N(\mathbf{w}; \mathbf{0}, \alpha^{-1} I) \tag{7.3}$$

因此参数后验分布

$$p(\mathbf{w}|D) = \frac{1}{Z}p(\mathbf{w}|\alpha)\prod_{i=1}^{N}p(y^{(i)}|\mathbf{x}^{(i)},\mathbf{w})$$
$$=\frac{1}{Z}p(\mathbf{w}|\alpha)\prod_{i=1}^{N}p(y^{(i)}|\mathbf{x}^{(i)},\mathbf{w})$$

其中 Z 是与 $\mathbf{w}$ 无关的归一化因子。最大化后验分布，相当于最小化其负对数。可以看到，和最大似然相比，只是多了和先验有关的一项

$$\max_{\mathbf{w}} -\log p(\mathbf{w}|D)$$
$$=\max_{\mathbf{w}} -\log\prod_{i=1}^{N}p(y^{(i)}|\mathbf{x}^{(i)},\mathbf{w}) - \log p(\mathbf{w}|\alpha)$$
$$=\max_{\mathbf{w}} \frac{\beta}{2}\sum_{i=1}^{N}(y^{(i)}-\mathbf{w}^T\mathbf{x}^{(i)})^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}$$

可以看到，该损失恰为岭回归的损失，先验相关的一项恰好对应 L2 正则化。正则化时使用的超参数 α 实际上对应参数高斯先验的精度。因此，这一优化问题有闭式解

$$\mathbf{w}^\star = (X^TX+\lambda I)^{-1}X^T\mathbf{y} \tag{7.4}$$

7.2.3 后验分布

在贝叶斯方法中，仅仅估计一个最优参数值是不够的，还要求出具体的后验分布。

算法	**线性回归-后验分布**
算法简述	对于回归问题，假设标签服从高斯分布，其均值为样本的线性变换， 参数服从高斯先验，求出后验分布
已知	样本 $X \in R^{n\times N}$（扩充过），标签 $Y \in R^N$，先验 $p(\mathbf{w}\|\alpha) = N(\mathbf{w};\mathbf{0},\alpha^{-1}I)$
求	线性系数 $\mathbf{w} \in R^n$ 的后验分布 $p(\mathbf{w}\|D) = N(\mathbf{w};\mu_\mathbf{w},\Sigma_\mathbf{w})$
解类型	闭式解
闭式解	方差 $\Sigma_\mathbf{w} = \alpha I + \beta XX^T$ 均值 $\mu_\mathbf{w} = (\alpha I + \beta XX^T)^{-1}XY$

我们知道，对于均值的参数而言，高斯分布的先验是共轭先验，即后验分布也有高斯分布的形式。不妨设

$$p(\mathbf{w}|D) = N(\mathbf{w};\mu_\mathbf{w},\Sigma_\mathbf{w}) \tag{7.5}$$

由

$$p(\mathbf{w}|D) \propto p(\mathbf{w}|\alpha)\prod_{i=1}^{N}p(y^{(i)}|\mathbf{x}^{(i)},\mathbf{w})$$

有

$$\exp\{-\frac{1}{2}(\mathbf{w}-\mu_{\mathbf{w}})^T\Sigma_{\mathbf{w}}^{-1}(\mathbf{w}-\mu_{\mathbf{w}})\} \propto \exp\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\}\prod_{i=1}^{N}\exp\{-\frac{\beta}{2}(y^{(i)}-\mathbf{w}^T\mathbf{x}^{(i)})^2\}$$

即

$$(\mathbf{w}-\mu_{\mathbf{w}})^T\Sigma_{\mathbf{w}}^{-1}(\mathbf{w}-\mu_{\mathbf{w}}) \propto \alpha\mathbf{w}^T\mathbf{w}+\beta\sum_{i=1}^{N}(y^{(i)}-\mathbf{w}^T\mathbf{x}^{(i)})^2$$

分别对比二次项和一次项有

$$\begin{aligned}\mathbf{w}^T\Sigma_{\mathbf{w}}\mathbf{w} &\propto \alpha\mathbf{w}^T\mathbf{w}+\beta\mathbf{w}^TXX^T\mathbf{w}\\ \mu_{\mathbf{w}}^T\Sigma_{\mathbf{w}}^{-1}\mathbf{w} &\propto \beta Y^TX^T\mathbf{w}\end{aligned}$$

因此

$$\begin{aligned}\mu_{\mathbf{w}} =&\beta(\alpha I+\beta XX^T)^{-1}XY\\ \Sigma_{\mathbf{w}} =&(\alpha I+\beta XX^T)^{-1}\end{aligned} \tag{7.6}$$

7.2.4 预测模型

算法	**线性回归-预测模型**
算法简述	对于回归问题，假设标签服从高斯分布，其均值为样本的线性变换， 参数服从高斯先验，求标签的预测模型
已知	样本 $X\in R^{n\times N}$（扩充过），标签 $Y\in R^N$，先验 $p(\mathbf{w}\|\alpha)=N(\mathbf{w};\mathbf{0},\alpha^{-1}I)$
求	新样本 $\mathbf{x}$ 条件下标签的分布 $p(y\|\mathbf{x})$
解类型	闭式解
闭式解	方差 $\Sigma_y=\beta^{-1}+\mathbf{x}^T(\alpha I+\beta XX^T)^{-1}\mathbf{x}$ 均值 $\mu_y=\beta\mathbf{x}^T(\alpha I+\beta XX^T)^{-1}XY$

在使用学习到的模型进行预测时，要对参数在参数空间上进行积分

$$\begin{aligned}p(y|\mathbf{x},D) =& \int p(y|\mathbf{x},\mathbf{w},D)p(\mathbf{w}|D)d\mathbf{w}\\ =&\int p(y,\mathbf{w}|\mathbf{x},D)d\mathbf{w}\\ =&\frac{1}{Z}\int\exp\{-\beta(y-\mathbf{w}^T\mathbf{x})^2-(\mathbf{w}-\mu_{\mathbf{w}})^T\Sigma_{\mathbf{w}}^{-1}(\mathbf{w}-\mu_{\mathbf{w}})\}d\mathbf{w}\end{aligned}$$

即联合概率分布

$$p(y,\mathbf{w}|\mathbf{x},D)\propto\exp\{-\beta(y-\mathbf{w}^T\mathbf{x})^2-(\mathbf{w}-\mu_{\mathbf{w}})^T\Sigma_{\mathbf{w}}^{-1}(\mathbf{w}-\mu_{\mathbf{w}})\}$$

记

$$\mathbf{z}=\begin{pmatrix}\mathbf{w}\\ y\end{pmatrix}$$

则有

$$\begin{aligned}\log p(\mathbf{z}|\mathbf{x}, D) &\propto -\frac{\beta}{2}(y - \mathbf{w}^T\mathbf{x})^2 - \frac{1}{2}(\mathbf{w} - \mu_\mathbf{w})^T\Sigma_\mathbf{w}^{-1}(\mathbf{w} - \mu_\mathbf{w}) \\ &\propto -\frac{1}{2}(\beta y^2 + \mathbf{w}^T(\Sigma_\mathbf{w}^{-1} + \beta\mathbf{x}\mathbf{x}^T)\mathbf{w} - 2\beta y\mathbf{x}^T\mathbf{w} - 2\mu_\mathbf{w}^T\Sigma_\mathbf{w}^{-1}\mathbf{w})\end{aligned}$$

考虑其中关于 $\mathbf{w}$ y 的二次项，它们是和 $\mathbf{z}$ 协方差 Σ_z 相关的项

$$\begin{aligned}\mathbf{z}^T\Sigma_z^{-1}\mathbf{z} &= -\frac{1}{2}(\beta y^2 + \mathbf{w}^T(\Sigma_\mathbf{w}^{-1} + \beta\mathbf{x}\mathbf{x}^T)\mathbf{w} - 2\beta y\mathbf{x}^T\mathbf{w}) \\ &= \begin{pmatrix}\mathbf{w}^T & y\end{pmatrix}\begin{bmatrix}\Sigma_\mathbf{w}^{-1} + \beta\mathbf{x}\mathbf{x}^T & -y\mathbf{x} \\ -y\mathbf{x}^T & \beta\end{bmatrix}\begin{pmatrix}\mathbf{w} \\ y\end{pmatrix}\end{aligned}$$

因此

$$\Sigma_z = \begin{bmatrix}\Sigma_\mathbf{w}^{-1} + \beta\mathbf{x}\mathbf{x}^T & -y\mathbf{x} \\ -y\mathbf{x}^T & \beta\end{bmatrix}^{-1} = \begin{bmatrix}\Sigma_\mathbf{w} & \Sigma_\mathbf{w}\mathbf{x} \\ \mathbf{x}^T\Sigma_\mathbf{w} & \beta^{-1} + \mathbf{x}^T\Sigma_\mathbf{w}\mathbf{x}\end{bmatrix}$$

同样的，考虑一次项，它们是和均值 μ_z 相关的项

$$\mu_z^T\Sigma_z^{-1}\mathbf{z} = \mu_\mathbf{w}^T\Sigma_\mathbf{w}^{-1}\mathbf{w} = \begin{pmatrix}\mu_\mathbf{w}^T\Sigma_\mathbf{w}^{-1} & 0\end{pmatrix}\begin{pmatrix}\mathbf{w} \\ y\end{pmatrix}$$

因此

$$\begin{aligned}\Sigma_z^{-1}\mu_z &= \begin{pmatrix}\Sigma_\mathbf{w}^{-1}\mu_\mathbf{w} \\ 0\end{pmatrix} \\ \mu_z &= \begin{bmatrix}\Sigma_\mathbf{w} & \Sigma_\mathbf{w}\mathbf{x} \\ \mathbf{x}^T\Sigma_\mathbf{w} & \beta^{-1} + \mathbf{x}^T\Sigma_\mathbf{w}\mathbf{x}\end{bmatrix}\begin{pmatrix}\Sigma_\mathbf{w}^{-1}\mu_\mathbf{w} \\ 0\end{pmatrix} \\ \mu_z &= \begin{pmatrix}\mu_\mathbf{w} \\ \mathbf{x}^T\mu_\mathbf{w}\end{pmatrix}\end{aligned}$$

联合高斯分布的边缘概率的均值和协方差，就是联合分布的均值和方差中对应的部分。因此我们可以直接写出

$$\begin{aligned}\mu_y =& \mathbf{x}^T\mu_\mathbf{w} = \beta\mathbf{x}^T(\alpha I + \beta XX^T)^{-1}XY \\ \Sigma_y =& \beta^{-1} + \mathbf{x}^T\Sigma_\mathbf{w}\mathbf{x} = \beta^{-1} + \mathbf{x}^T(\alpha I + \beta XX^T)^{-1}\mathbf{x}\end{aligned} \tag{7.7}$$

预测模型为

$$p(y|\mathbf{x}, D) = N(y; \mu_y, \Sigma_y) \tag{7.8}$$

7.2.5 超参数证据近似

算法	线性回归-超参数证据近似
算法简述	对于线性回归的超参数估计，使用最大化证据函数的迭代算法估计超参数
已知	样本 $X \in R^{n \times N}$（扩充过），标签 $Y \in R^N$ 标签服从方差为 β^{-1} 高斯分布，回归系数服从方差为 $\alpha^{-1}I$ 高斯分布
求	超参数 α, β 的最优估计
解类型	迭代解
算法步骤	1. 随机初始化 α, β 2. 求出 XX^T 的 n 个特征值 $\lambda_1, \lambda_2, ..., \lambda_n$ 3. 使用以下迭代式迭代求解 $\gamma(t+1) = \sum_{i=1}^{n} \frac{\beta(t)\lambda_i}{\alpha(t) + \beta(t)\lambda_i}$ $\mu_w(t+1) = \beta(t)(\alpha(t)I + \beta(t)XX^T)^{-1}XY$ $\alpha(t+1) = \frac{\gamma(t+1)}{\mu_w(t+1)^T\mu_w(t+1)}$, $\beta(t+1) = \frac{N - \gamma(t+1)}{\|\|Y - X^T\mu_w\|\|^2}$

对于概率模型，我们评估模型参数的方法是计算似然函数

$$p(Y|\mathbf{w}, X) = \prod_{i=1}^{N} p(y^{(i)}|\mathbf{w}, \mathbf{x}^{(i)})$$

不过，这个函数其实是在假定超参数 α 和 β 是已知的情况下计算的。如果从贝叶斯学派观点来看，超参数也服从一定的分布，应当被视为随机变量。因此，我们其实计算的是 $p(Y|\mathbf{w}, X, \alpha, \beta)$。

既然是随机变量，就可以计算似然。超参数的似然表达式应当和参数 $\mathbf{w}$ 无关，即 $p(Y|\alpha, \beta, X)$。为了得到这个式子，应当对标签 Y 和参数 $\mathbf{w}$ 的联合（条件）分布进行积分

$$\begin{aligned} & p(Y|\alpha, \beta, X) \\ = & \int p(Y, \mathbf{w}|X, \alpha, \beta) d\mathbf{w} \\ = & \int p(Y|\mathbf{w}, X, \alpha, \beta) p(\mathbf{w}|X, \alpha, \beta) d\mathbf{w} \\ = & \int p(Y|\mathbf{w}, X, \alpha, \beta) p(\mathbf{w}|\alpha) d\mathbf{w} \end{aligned}$$

这个似然函数也被称为**证据函数**。同理，对于后验分布和预测模型，如果将超参数视为随机变量，也应当对其进行积分

$$p(Y|\mathbf{w}, X) = \int\int p(Y|\mathbf{w}, X, \alpha, \beta) p(\alpha|X) p(\beta|X) d\alpha d\beta$$

$$p(y|\mathbf{x}, D) = \int\int p(y|\mathbf{x}, D, \alpha, \beta) p(\alpha|D) p(\beta|D) d\alpha d\beta$$

然而，很多时候对超参数分布积分得到的边缘分布没有解析解。因此我们退而求其次，使用最大化证据函数的超参数作为积分的近似。这实际上是假设了超参数的后验分布接近于狄拉克分布 $\delta(\alpha)$。即

$$p(y|\mathbf{x}, D) \approx p(y|\mathbf{x}, D, \hat{\alpha}, \hat{\beta})$$

$$\hat{\alpha}, \hat{\beta} = \arg\max_{\alpha,\beta} p(Y|\alpha, \beta, X)$$

于是问题转化成了针对超参数的优化问题。最大化证据相当于最大化对数证据

$$\begin{aligned}
\max_{\alpha,\beta} L(\alpha, \beta) &= \log p(Y|\alpha, \beta, X) \\
&= \log \int p(Y, \mathbf{w}|X, \alpha, \beta) d\mathbf{w} \\
&= \log \int p(Y|\mathbf{w}, X, \beta) p(\mathbf{w}|\alpha) d\mathbf{w} \\
&= \log \int \prod_{i=1}^{N} p(y^{(i)}|\mathbf{w}, \mathbf{x}^{(i)}, \beta) p(\mathbf{w}|\alpha) d\mathbf{w}
\end{aligned}$$

其中

$$\begin{aligned}
& p(Y, \mathbf{w}|X, \alpha, \beta) \\
=& \prod_{i=1}^{N} p(y^{(i)}|\mathbf{w}, \mathbf{x}^{(i)}, \beta) \\
=& \prod_{i=1}^{N} \left(\sqrt{\frac{\beta}{(2\pi)^n}} \exp\{-\frac{\beta}{2}(y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2\} \right) \\
=& \left(\frac{\beta}{(2\pi)^n} \right)^{\frac{N}{2}} \exp\{-\frac{\beta}{2} \sum_{i=1}^{N} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2\} \\
=& \left(\frac{\beta}{(2\pi)^n} \right)^{\frac{N}{2}} \exp\{-\frac{\beta}{2}(Y^T Y - 2Y^T X^T \mathbf{w} + \mathbf{w}^T X X^T \mathbf{w})\}
\end{aligned}$$

而

$$p(\mathbf{w}|\alpha) = \left(\frac{\alpha}{(2\pi)} \right)^{\frac{n}{2}} \exp\{-\frac{\alpha}{2} \mathbf{w}^T \mathbf{w}\}$$

因此

$$\begin{aligned}
& p(Y, \mathbf{w}|X, \alpha, \beta) \\
=& \left(\frac{\beta}{(2\pi)^n} \right)^{\frac{N}{2}} \left(\frac{\alpha}{2\pi} \right)^{\frac{n}{2}} \exp\{-\frac{\beta}{2}(Y^T Y - 2Y^T X^T \mathbf{w} + \mathbf{w}^T X X^T \mathbf{w}) - \frac{\alpha}{2} \mathbf{w}^T \mathbf{w}\} \\
=& \left(\frac{\beta}{(2\pi)^n} \right)^{\frac{N}{2}} \left(\frac{\alpha}{2\pi} \right)^{\frac{n}{2}} \exp\{-\frac{1}{2}(\mathbf{w}^T(\alpha I + \beta X X^T)\mathbf{w} - 2\beta Y^T X^T \mathbf{w} + \beta Y^T Y)\} \\
=& F(\alpha, \beta) \exp\{-E(\mathbf{w})\}
\end{aligned} \tag{7.9}$$

即

$$L(\alpha, \beta) = \log \int F(\alpha, \beta) \exp\{-E(\mathbf{w})\} d\mathbf{w} \tag{7.10}$$

其中

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{w}^T(\alpha I + \beta X X^T)\mathbf{w} - 2\beta Y^T X^T \mathbf{w} + \beta Y^T Y) \tag{7.11}$$

$$F(\alpha,\beta)=\left(\frac{\beta}{(2\pi)^n}\right)^{\frac{N}{2}}\left(\frac{\alpha}{2\pi}\right)^{\frac{n}{2}} \tag{7.12}$$

$E(\mathbf{w})$ 是关于 $\mathbf{w}$ 的二次表达式。我们在之前已经使用配方计算过，即式 7.6的结果。

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}(\mathbf{w}-\mu_w)^T\Sigma_w^{-1}(\mathbf{w}-\mu_w)+E(\mu_w) \\ \Sigma_w &= (\alpha I+\beta XX^T)^{-1} \\ \mu_w &= \beta(\alpha I+\beta XX^T)^{-1}XY \end{aligned} \tag{7.13}$$

只不过这里还有一些和 μ_w 无关的余项，我们记为 $E(\mu_w)$，因为

$$\begin{aligned} E(\mu_w) =& \frac{\beta}{2}Y^TY-\frac{1}{2}\mu_w^T\Sigma_w^{-1}\mu_w \\ =& \frac{1}{2}(\beta Y^TY-2\mu_w^T\Sigma_w^{-1}\mu_w)+\frac{1}{2}\mu_w^T\Sigma_w^{-1}\mu_w \\ =& \frac{\beta}{2}(Y^TY-2\mu_w^TXY)+\frac{1}{2}\mu_w^T\Sigma_w^{-1}\mu_w \\ =& \frac{\beta}{2}||Y-X^T\mu_w||^2-\frac{\beta}{2}\mu_w^TXX^T\mu_w+\frac{1}{2}\mu_w^T\Sigma_w^{-1}\mu_w \\ =& \frac{\beta}{2}||Y-X^T\mu_w||^2+\frac{\alpha}{2}\mu_w^T\mu_w \end{aligned}$$

此时可以利用高斯分布性质计算积分

$$\begin{aligned} L(\alpha,\beta) =& \log\int F(\alpha,\beta)\exp\{-E(\mathbf{w})\}d\mathbf{w} \\ =& \log\left(F(\alpha,\beta)\exp\{-E(\mu_w)\}\int\exp\{-\frac{1}{2}(\mathbf{w}-\mu_w)^T\Sigma_w^{-1}(\mathbf{w}-\mu_w)\}d\mathbf{w}\right) \\ =& \log F(\alpha,\beta)-E(\mu_w)+\log\sqrt{(2\pi)^n|\Sigma_w|} \end{aligned}$$

其中

$$\log F(\alpha,\beta)=\frac{N}{2}(\log\beta-n\log 2\pi)+\frac{n}{2}(\log\alpha-\log 2\pi)$$

$$\log\sqrt{(2\pi)^n|\Sigma_w|}=\frac{n}{2}\log 2\pi+\frac{1}{2}\log|\Sigma_w|$$

因此

$$L(\alpha,\beta)=-\frac{\alpha}{2}\mu_w^T\mu_w+\frac{n}{2}\log\alpha+\frac{N}{2}\log\beta-\frac{\beta}{2}||Y-X^T\mu_w||^2+\frac{1}{2}\log|\Sigma_w|+const$$

这里 μ_w 对 α,β 的依赖较为复杂，在求导时先忽略。求偏导有

$$\begin{aligned} \frac{\partial L}{\partial\alpha} &= -\frac{1}{2}\mu_w^T\mu_w+\frac{n}{2\alpha}+\frac{1}{2}\frac{\partial}{\partial\alpha}\log|\Sigma_w| \\ \frac{\partial L}{\partial\beta} &= -\frac{1}{2}||Y-X^T\mu_w||^2+\frac{N}{2\beta}+\frac{1}{2}\frac{\partial}{\partial\beta}\log|\Sigma_w| \end{aligned}$$

假设 XX^T 的相似对角化为

$$XX^T=P^T\Lambda P$$

其中 Λ 是对角阵，对应 XX^T 的特征值 $\lambda_1, \lambda_2, ..., \lambda_n$。因此有

$$\begin{aligned}|\alpha I + \beta XX^T| &= |\alpha I + \beta P^T \Lambda P| = |P^T||\alpha I + \beta \Lambda||P| \\ &= |\alpha I + \beta \Lambda| = \prod_{i=1}^{n}(\alpha + \beta\lambda_i)\end{aligned}$$

即

$$\log|\Sigma_w| = \log|(\alpha I + \beta XX^T)^{-1}| = -\log|\alpha I + \beta XX^T| = -\sum_{i=1}^{n}\log(\alpha + \beta\lambda_i)$$

因此有关于 α 的梯度

$$\frac{\partial}{\partial\alpha}\log|\Sigma_w| = -\sum_{i=1}^{n}\frac{1}{\alpha + \beta\lambda_i}$$

以及关于 β 的梯度

$$\frac{\partial}{\partial\beta}\log|\Sigma_w| = -\sum_{i=1}^{n}\frac{\lambda_i}{\alpha + \beta\lambda_i}$$

因此

$$\begin{aligned}\frac{\partial L}{\partial\alpha} &= -\frac{1}{2}\mu_w^T\mu_w + \frac{n}{2\alpha} - \frac{1}{2}\sum_{i=1}^{n}\frac{1}{\alpha + \beta\lambda_i} \\ \frac{\partial L}{\partial\beta} &= -\frac{1}{2}||Y - X^T\mu_w||^2 + \frac{N}{2\beta} - \frac{1}{2}\sum_{i=1}^{n}\frac{\lambda_i}{\alpha + \beta\lambda_i}\end{aligned}$$

令偏导为 0，有

$$n - \alpha\sum_{i=1}^{n}\frac{1}{\alpha + \beta\lambda_i} = \alpha\mu_w^T\mu_w$$

$$\sum_{i=1}^{n}(1 - \frac{\alpha}{\alpha + \beta\lambda_i}) = \alpha\mu_w^T\mu_w$$

$$\alpha = \frac{1}{\mu_w^T\mu_w}\sum_{i=1}^{n}\frac{\beta\lambda_i}{\alpha + \beta\lambda_i}$$

定义

$$\gamma(\alpha, \beta) = \sum_{i=1}^{n}\frac{\beta\lambda_i}{\alpha + \beta\lambda_i}$$

则有

$$\alpha = \frac{\gamma(\alpha, \beta)}{\mu_w^T\mu_w}$$

以及

$$\frac{N}{\beta} - \sum_{i=1}^{n}\frac{\lambda_i}{\alpha + \beta\lambda_i} = ||Y - X^T\mu_w||^2$$

$$\frac{N}{\beta}-\frac{\gamma(\alpha,\beta)}{\beta}=||Y-X^T\mu_w||^2$$

$$\beta=\frac{N-\gamma(\alpha,\beta)}{||Y-X^T\mu_w||^2}$$

由此我们得到了最优解的迭代公式

$$\begin{aligned}\gamma&=\sum_{i=1}^{n}\frac{\beta\lambda_i}{\alpha+\beta\lambda_i}\\ \mu_w&=\beta(\alpha I+\beta XX^T)^{-1}XY\\ \alpha&=\frac{\gamma}{\mu_w^T\mu_w}\\ \beta&=\frac{N-\gamma}{||Y-X^T\mu_w||^2}\end{aligned}\tag{7.14}$$

只需要任意初始化 α,β，迭代至收敛即可求出最优解 $\hat{\alpha},\hat{\beta}$

7.2.6 超参数 E-M 算法

算法	**线性回归-超参数 EM 算法**
算法简述	对于线性回归的超参数估计，认为参数是隐变量，使用 E-M 算法迭代估计超参数
已知	样本 $X\in R^{n\times N}$（扩充过），标签 $Y\in R^N$ 标签服从方差为 β^{-1} 高斯分布，回归系数服从方差为 $\alpha^{-1}I$ 高斯分布
求	超参数 α,β 的最优估计
解类型	迭代解
算法步骤	1. 随机初始化 α,β 2. 使用以下迭代式迭代求解 $\mu_w(t+1)=\beta(t)(\alpha(t)I+\beta(t)XX^T)^{-1}XY$ $\alpha(t+1)=\frac{n}{\mu_w(t+1)^T\mu_w(t+1)}$ $\beta(t+1)=\frac{N}{\|Y-X^T\mu_w(t+1)\|^2}$

对于线性回归的隐变量最大证据问题，我们也可以从 E-M 算法的角度来解。在这里，我们认为待优化参数是 α,β，而隐变量是线性回归的回归系数 $\mathbf{w}$。我们有隐变量后验分布

$$p(\mathbf{w}|Y,\alpha,\beta,X)=N(\mathbf{w};\mu_w,\Sigma_w)$$

其中 μ_w 和 Σ_w 由式7.6给出。另一方面，由式7.9，我们有联合分布对数

$$\log p(Y,\mathbf{w}|\alpha,\beta,X)=\log p(Y|\mathbf{w},X,\beta,\alpha)+\log p(\mathbf{w}|\alpha)=\log F(\alpha,\beta)-E(\mathbf{w})$$

其中 $F(\alpha,\beta)$ 和 $E(\mathbf{w})$ 分别由式7.13和7.12给出。在 E 步，我们需要计算联合分布对数对隐变量后验的期望，即

$$
\begin{aligned}
LB(\alpha,\beta) =& \big\langle \log p(Y,\mathbf{w}|\alpha,\beta,X)\big\rangle_{p(\mathbf{w}|Y,\alpha,\beta)} \\
=& \big\langle \log F(\alpha,\beta) - E(\mathbf{w})\big\rangle_{p(\mathbf{w}|Y,\alpha,\beta,X)} \\
=& \log F(\alpha,\beta) - E(\mu_w) - \frac{1}{2}\big\langle (\mathbf{w}-\mu_w)^T\Sigma_w^{-1}(\mathbf{w}-\mu_w)\big\rangle_{N(\mathbf{w};\mu_w,\Sigma_w)} \\
=& \frac{N}{2}(\log\beta - n\log 2\pi) + \frac{n}{2}(\log\alpha - \log 2\pi) - \frac{\beta}{2}||Y-X^T\mu_w||^2 - \frac{\alpha}{2}\mu_w^T\mu_w - \frac{n}{2} \\
=& \frac{N}{2}\log\beta + \frac{n}{2}\log\alpha - \frac{\beta}{2}||Y-X^T\mu_w||^2 - \frac{\alpha}{2}\mu_w^T\mu_w + const
\end{aligned}
$$

这里的证明使用了结论2.41。

对于 E 步，我们根据式7.13计算 μ_w。即：

$$
\mu_w(t+1) = \beta(t)(\alpha(t)I + \beta(t)XX^T)^{-1}XY \tag{7.15}
$$

对于 M 步，我们不考虑 μ_w 和 α,β 的关系，进行最大化。进行求导有

$$
\frac{\partial LB}{\partial\alpha} = \frac{n}{2\alpha} - \frac{1}{2}\mu_w^T\mu_w = 0
$$

$$
\frac{\partial LB}{\partial\beta} = \frac{N}{2\beta} - \frac{1}{2}||Y-X^T\mu_w||^2 = 0
$$

因此 M 步迭代式为

$$
\begin{aligned}
\alpha(t+1) &= \frac{n}{\mu_w(t+1)^T\mu_w(t+1)} \\
\beta(t+1) &= \frac{N}{||Y-X^T\mu_w(t+1)||^2}
\end{aligned} \tag{7.16}
$$

这样的表达式和证据近似算法的结果稍有不同，不过仍然可以收敛到相同的解。

重要提示

读者可能会发现，本小节给出的更新公式和 PRML 一书 9.3.4 节中不同。事实上，这是因为两者推导的出发点不同导致的。在我们的推导中我们认为

$$
p(Y,\mathbf{w}|\alpha,\beta,X) = p(Y|\mathbf{w},\alpha,\beta,X)p(\mathbf{w}|\alpha)
$$

而 PRML 中给出的公式是

$$
p(Y,\mathbf{w}|\alpha,\beta,X) = p(Y|\mathbf{w},\beta,X)p(\mathbf{w}|\alpha)
$$

这一点微小的区别来源于对概率模型的不同认识。$p(Y|\mathbf{w},\beta,X)$ 的表达式意味着参数 $\mathbf{w}$ 没有先验分布，或其先验分布为一个均匀分布（相当于 $\alpha\to 0$，于是 $p(\mathbf{w}|\alpha)$ 的方差 $\alpha^{-1}I$ 趋近于无穷大）。而 $p(Y|\mathbf{w},\beta,\alpha,X)$ 意味着参数 $\mathbf{w}$ 由 α 控制先验分布。

我们本小节的目的是计算超参数 α,β 的最佳值，但 $\mathbf{w}$ 的取值也会影响超参数 α,β 的似然。因此我们将其视为隐变量，使用 E-M 算法。既然 $\mathbf{w}$ 是隐变量，我们自然应当考虑它和“参数” α,β 的**全部**关联。因此，作者谨认为，PRML 中的处理是不妥的。

关于超参数的最优估计，有上小节的证据近似、本小节的 E-M 算法。如果再加上 PRML 一书中的推导就是三种算法。读者可以自行编写程序，验证这三种算法究竟能不能收敛到相同的结果。

7.3 概率 Logistic 回归

Logistic 回归也已经在第 3 章中介绍过，但概率方法有着不同的视角。

对于二分类任务，从概率的角度，可以认为样本的标签 y 是一个 0-1 随机变量，服从参数为 p 的伯努利分布，而参数 p 由样本 $\mathbf{x}$ 和参数 $\mathbf{w}$ 控制，即 $y \sim p(y|\mathbf{x}, \mathbf{w}) = B(f(\mathbf{x}, \mathbf{w}))$。由于 p 限定在 $[0, 1]$ 中，因此考虑取 f 的形式为 sigmoid 函数

$$p = f(\mathbf{x}, \mathbf{w}) = \text{sig}(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp\{-\mathbf{w}^T\mathbf{x}\}}$$

这样我们建立了一个判别模型。根据伯努利分布定义，写出标签的后验概率

$$p(y|\mathbf{x}, \mathbf{w}) = (\text{sig}(\mathbf{w}^T\mathbf{x}))^y(1 - \text{sig}(\mathbf{w}^T\mathbf{x}))^{1-y}$$

现在问题就变成，如何根据数据集 $X \in R^{n\times N}$, $Y = \{y^{(i)} \in \{0, 1\}, i = 1, 2, ..., N\}$, $D = \{X, Y\}$ 求出参数。

7.3.1 最大后验估计

算法	**Logistic 回归-最大后验**
算法简述	对于二分类问题，假设样本服从 Logistic 线性模型控制的伯努利分布，线性系数服从高斯先验分布，求 MAP 准则下的参数点估计
已知	样本 $X \in R^{n\times N}$（扩充过），标签 $y^{(i)} \in \{0, 1\}$，学习率 λ
求	$\mathbf{w} \in R^n$ 使得 $\max_{\mathbf{w}} p(\mathbf{w}|D)$
解类型	迭代解
迭代式	$\mathbf{w}(t+1) = \mathbf{w}(t) + \lambda(\frac{1}{N}\sum_{i=1}^{N}(\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - y^{(i)})\mathbf{x}^{(i)} + \alpha\mathbf{w}(t))$

和线性回归一样，我们考虑最大化后验概率，即求 $\mathbf{w}_{MAP}$。假设 $\mathbf{w}$ 的先验分布是均值为 $\mathbf{0}$、方差为 $\alpha^{-1}I$ 的高斯分布

$$p(\mathbf{w}|\alpha) = N(w; \mathbf{0}, \alpha^{-1}I) = \left(\frac{\alpha}{2\pi}\right)^{\frac{n}{2}} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right\} \tag{7.17}$$

其中 α 称为精度，是一个超参数，需要人为提前给定。$\mathbf{w}$ 的似然函数即标签的后验概率

$$p(Y|X, \mathbf{w}) = \prod_{i=1}^{N}(\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))^{y^{(i)}}(1 - \text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))^{1-y^{(i)}} \tag{7.18}$$

则参数后验

$$p(\mathbf{w}|D) = \frac{p(Y|X, \mathbf{w})p(\mathbf{w}|\alpha)}{\int_{\mathbf{w}} p(Y|X, \mathbf{w})p(\mathbf{w}|\alpha)}$$

最大化后验概率相当于最小化负对数。由于分母与 $\mathbf{w}$ 无关，因此仅考虑分子部分

$$\max_{\mathbf{w}} p(\mathbf{w}|D) = \min_{\mathbf{w}} -\log p(\mathbf{w}|D) = \min_{\mathbf{w}} -\log p(Y|X, \mathbf{w}) - \log p(\mathbf{w}|\alpha)$$

进行展开

$$
\begin{aligned}
\min_{\mathbf{w}} L(\mathbf{w}) &= -\log p(Y|X,\mathbf{w}) - \log p(\mathbf{w}|\alpha) \\
&= -\log \prod_{i=1}^{N} (\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))^{y^{(i)}} (1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))^{1-y^{(i)}} - \log N(w;\mathbf{0},\alpha I) \\
&= -\sum_{i=1}^{N} (y^{(i)} \log \mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) + (1-y^{(i)}) \log(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))) - \frac{n}{2}(\log\alpha - \log 2\pi) + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} \\
&= -\sum_{i=1}^{N} y^{(i)} \log \mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - \sum_{i=1}^{N} (1-y^{(i)}) \log(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})) + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}
\end{aligned}
$$

可以看到，$\mathbf{w}$ 的高斯先验分布相当于对参数进行 L2 正则化。

求偏导

$$
\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}} &= -\sum_{i=1}^{N} y^{(i)} \frac{\partial}{\partial \mathbf{w}} \log \mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - \sum_{i=1}^{N} (1-y^{(i)}) \frac{\partial}{\partial \mathbf{w}} \log(1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})) + \alpha\mathbf{w} \\
&= -\sum_{i=1}^{N} y^{(i)} (1-\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))\mathbf{x}^{(i)} + \sum_{i=1}^{N} (1-y^{(i)}) \mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)})\mathbf{x}^{(i)} + \alpha\mathbf{w} \\
&= \sum_{i=1}^{N} (\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - y^{(i)})\mathbf{x}^{(i)} + \alpha\mathbf{w}
\end{aligned}
$$

可以看到，MAP 的梯度表达式分为回归损失和正则化损失两部分。回归损失和之前使用交叉熵损失进行推导的结果3.6完全一样。正则化部分仍然是对应了岭回归的 L2 正则化.

写出参数迭代更新式

$$
\mathbf{w}(t+1) = \mathbf{w}(t) + \lambda(\frac{1}{N}\sum_{i=1}^{N} (\mathrm{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) - y^{(i)})\mathbf{x}^{(i)} + \alpha\mathbf{w}(t)) \tag{7.19}
$$

7.3.2 拉普拉斯近似

算法	**Logistic 回归-拉普拉斯近似**
算法简述	对于二分类问题，假设样本服从 Logistic 线性模型控制的伯努利分布，线性系数服从高斯先验分布，求高斯近似的后验分布
已知	样本 $X \in R^{n\times N}$（扩充过），标签 $y^{(i)} \in \{0,1\}$
求	参数近似后验分布 $\tilde{p}(\mathbf{w}|D)$
解类型	近似解
近似解	$\tilde{p}(\mathbf{w}|D) = N(\mathbf{w};\mathbf{w}_{MAP},\Sigma_L)$

除了求出最大后验的参数点估计外，还应该求出参数的后验概率。然而，Logistic 回归中似然函数关于参数 $\mathbf{w}$ 的形式比较复杂，难以求出共轭先验。例如，采用高斯先验分布

$$
p(Y|X,\mathbf{w}) = \prod_{i=1}^{N} \mathrm{sig}(y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}) \tag{7.20}
$$

那么后验概率为

$$p(\mathbf{w}|D) = \frac{p(\mathbf{w}|\alpha)\prod_{i=1}^{N}(\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))^{y^{(i)}}(1-\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))^{1-y^{(i)}}}{\int_{\mathbf{w}} p(\mathbf{w}|\alpha)\prod_{i=1}^{N}\text{sig}(y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)})}$$

这一概率分布十分复杂。因此使用拉普拉斯近似来求解一个近似分布。首先计算对数

$$\log p(\mathbf{w}|D) = \log p(\mathbf{w}|\alpha) + \sum_{i=1}^{N}\log(\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))^{y^{(i)}}(1-\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))^{1-y^{(i)}} + C$$

$$= \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N}(y^{(i)}\log\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}) + (1-y^{(i)})\log(1-\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))) + C$$

求导

$$\frac{\partial}{\partial\mathbf{w}}\log p(\mathbf{w}|D) = \sum_{i=1}^{N}(y^{(i)} - \text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))\mathbf{x}^{(i)} + \alpha\mathbf{w}$$

二阶导

$$\frac{\partial^2}{\partial\mathbf{w}^2}\log p(\mathbf{w}|D) = \sum_{i=1}^{N} -\frac{\partial}{\partial\mathbf{w}}\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)})\mathbf{x}^{(i)^T} + \alpha I$$

$$= \alpha I - \sum_{i=1}^{N}\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)})(1-\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))\mathbf{x}^{(i)}\mathbf{x}^{(i)^T}$$

设矩阵

$$\Sigma_L = -(\alpha I - \sum_{i=1}^{N}\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)})(1-\text{sig}(\mathbf{w}^T\mathbf{x}^{(i)}))\mathbf{x}^{(i)}\mathbf{x}^{(i)^T})^{-1} \tag{7.21}$$

则近似后验分布为

$$\tilde{p}(\mathbf{w}|D) = N(\mathbf{w}; \mathbf{w}_{MAP}, \Sigma_L) \tag{7.22}$$

7.4 最大熵模型

最大熵模型是一种用于分类任务的机器学习模型。它的提出来源于一个朴素的想法：最佳的概率分布估计是当前已知信息下熵最大的概率分布。

具体来说，我们希望我们提出的分类模型是一个判别式模型 $p(y|\mathbf{x})$，是一个条件分布。相应的，我们有条件熵的定义

$$H[p(y|\mathbf{x})] = -\int\left(\int p(y|\mathbf{x})\log p(y|\mathbf{x})dy\right)p(\mathbf{x})d\mathbf{x}$$

我们的目标就是在一定的条件下最大化条件熵。

同时，我们还有数据集 $D = \{X, \mathbf{y}\}$。我们希望模型尽量好的拟合数据，也就是

$$\tilde{p}(\mathbf{x})p(y|\mathbf{x}) = \tilde{p}(y, \mathbf{x}), \forall\mathbf{x}, y$$

这里 $\tilde{p}(\mathbf{x})$ 表示训练集样本的统计分布，$p(y|\mathbf{x})$ 是我们要求的模型，乘积表示用模型预测的样本和标签的联合分布。而 $\tilde{p}(y, \mathbf{x})$ 表示样本和标签联合分布的统计分布，即数据集中能得出的联合分布。如果模型预测和“实际”相同，那么说明模型能较好地拟合数据。

直接对这个等式进行约束是比较困难的，因为我们要遍历所有的 x, y 取值。为解决这个问题，我们可以借助在数据集上定义的特征函数 $f_i(\mathbf{x}, y), i = 1, 2, ..., K$。

特征函数是一类人为事先定义的函数，或者说是人为选定的数据特征。例如，如果样本 $\mathbf{x}$ 和标签 y 都是离散变量，取值个数分别为 P 和 Q，那么可以通过定义 $P \times Q$ 个二值函数遍历它们的每一种取值。实际中可以借助对数据集的先验知识涉及特征函数。为便于使用，特征函数一般取二值函数。

如果定义了 K 个特征函数，那么可以对每个特征函数在模型预测的联合分布和数据集统计的联合分布上分别求期望，并要求两个期望相等，即

$$\int\int f_i(\mathbf{x}, y)\tilde{p}(\mathbf{x})p(y|\mathbf{x})d\mathbf{x}dy = \int\int f_i(\mathbf{x}, y)\tilde{p}(y, \mathbf{x})d\mathbf{x}dy,\ i = 1, 2, ..., K$$

或

$$\langle f_i(\mathbf{x}, y)\rangle_{\tilde{p}(\mathbf{x})p(y|\mathbf{x})} = \langle f_i(\mathbf{x}, y)\rangle_{\tilde{p}(y,\mathbf{x})},\ i = 1, 2, ..., K$$

如果有 K 个特征函数，就能得到 K 个这样的约束条件。结合此前最大化条件熵的目标以及条件熵自身的归一化约束，我们可以得到这样一个约束最优化问题

$$\begin{aligned}\max_{p(y|x)} H[p(y|\mathbf{x})] = \quad & -\int\left(\int p(y|\mathbf{x})\log p(y|\mathbf{x})dy\right)\tilde{p}(\mathbf{x})d\mathbf{x} \\ \text{s.t.} \quad & \langle f_i(\mathbf{x}, y)\rangle_{\tilde{p}(\mathbf{x})p(y|\mathbf{x})} = \langle f_i(\mathbf{x}, y)\rangle_{\tilde{p}(y,\mathbf{x})},\ i = 1, 2, ..., K \\ & \int p(y|\mathbf{x})dy = 1\end{aligned}$$

按照最优化问题的惯例，我们将其写为最小化问题

$$\begin{aligned}\min_{p(y|x)} -H[p(y|\mathbf{x})] = \quad & \int\left(\int p(y|\mathbf{x})\log p(y|\mathbf{x})dy\right)\tilde{p}(\mathbf{x})d\mathbf{x} \\ \text{s.t.} \quad & \langle f_i(\mathbf{x}, y)\rangle_{\tilde{p}(\mathbf{x})p(y|\mathbf{x})} = \langle f_i(\mathbf{x}, y)\rangle_{\tilde{p}(y,\mathbf{x})},\ i = 1, 2, ..., K \\ & \int p(y|\mathbf{x})dy = 1\end{aligned} \tag{7.23}$$

这是一个约束最优化问题，约束条件复杂。不过，根据第2.2节介绍的最优化理论，我们可以将其化为相应的对偶问题7.24求解。由于问题对于优化变量（函数 $p(y|x)$）是凸的，原问题最优解等价于对偶问题最优解。

$$\max_{\lambda}\min_{p(y|x)} \mathcal{L}(p, \lambda) \tag{7.24}$$

其中

$$\begin{aligned}\mathcal{L}(p, \lambda) = & \int\left(\int p(y|\mathbf{x})\log p(y|\mathbf{x})dy\right)\tilde{p}(\mathbf{x})d\mathbf{x} + \lambda_0(1 - \int p(y|\mathbf{x})dy) \\ & + \sum_{i=1}^{K}\lambda_i(\langle f_i(\mathbf{x}, y)\rangle_{\tilde{p}(y,\mathbf{x})} - \langle f_i(\mathbf{x}, y)\rangle_{\tilde{p}(\mathbf{x})p(y|\mathbf{x})})\end{aligned} \tag{7.25}$$

为表达方便，在不引起歧义的情况下，我们将 $p(y|\mathbf{x})$ 缩写为 p。式7.24和7.25即为最大熵模型可以求解的优化问题。

算法	**最大熵-梯度上升法**
算法简述	对于分类问题，求条件熵最大的判别模型，通过对偶问题求解
已知	样本 $X \in R^{n\times N}$，标签 $Y=[\mathbf{y^{(1)}},\mathbf{y^{(2)}},...,\mathbf{y^{(N)}}] \in R^{c\times N}$， 特征函数 $f_1(\mathbf{x},y),...,f_K(\mathbf{x},y)$，学习率 α
求	判别模型 $p(y\|\mathbf{x},\lambda)$
解类型	迭代解
算法步骤	1. 随机初始化参数 $\lambda(0)\in R^K$ 2. 对每一样本 $\mathbf{x}^{(k)}$，计算当前参数下的标签条件分布 $Z_\lambda(\mathbf{x})=\int \exp\{\sum_{i=1}^K \lambda_i f_i(\mathbf{x},y)\}dy$ $p_\lambda(y\|\mathbf{x})=\frac{1}{Z_\lambda(\mathbf{x})}\exp\{\sum_{i=1}^K \lambda_i f_i(\mathbf{x},y)\}$ 3. 按照下式更新参数 $\lambda_i(t+1)=\lambda_i(t)+\alpha\sum_{k=1}^N \frac{1}{N}\left(f_i(\mathbf{x}^{(k)},y^{(k)})-\sum_y f_i(\mathbf{x}^{(k)},y)p_{\lambda(t)}(y\|\mathbf{x}^{(k)})\right)$ 4. $t\leftarrow t+1$, 若参数收敛，停止迭代，否则回到 2.

7.4.1 梯度上升法

可以看到，对偶问题的内层是一个较为简单的无约束优化问题。我们对条件概率求偏导，有

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial p}&=\int\left(\int(\log p(y|\mathbf{x})+1)dy\right)\tilde{p}(\mathbf{x})d\mathbf{x}-\lambda_0\int 1dy\\&\quad-\sum_{i=1}^K\lambda_i\int\int\tilde{p}(\mathbf{x})f_i(\mathbf{x},y)dyd\mathbf{x}\\&=\int\tilde{p}(\mathbf{x})\left(\int(\log p(y|\mathbf{x})+1-\lambda_0-\sum_{i=1}^K\lambda_i f_i(\mathbf{x},y))dy\right)d\mathbf{x}\end{aligned}$$

这里推导过程中用到了

$$\int 1dy=\int\int\tilde{p}(\mathbf{x})d\mathbf{x}dy$$

令

$$\frac{\partial\mathcal{L}}{\partial p}=0,\forall\mathbf{x},y$$

我们有

$$\log p(y|\mathbf{x})+1-\lambda_0-\sum_{i=1}^K\lambda_i f_i(\mathbf{x},y)=0,\ \forall\mathbf{x},y$$

因此解得

$$p(y|\mathbf{x})=\exp\{\sum_{i=1}^K\lambda_i f_i(\mathbf{x},y)\}\exp\{\lambda_0-1\}$$

由归一化约束，我们有

$$\int p(y|\mathbf{x})dy = \int \exp\{\sum_{i=1}^{K} \lambda_i f_i(\mathbf{x}, y)\}dy \exp\{\lambda_0 - 1\} = 1$$

设配分函数为

$$Z_\lambda(\mathbf{x}) = \int \exp\{\sum_{i=1}^{K} \lambda_i f_i(\mathbf{x}, y)\}dy \tag{7.26}$$

则有

$$\exp\{\lambda_0 - 1\} = \frac{1}{Z_\lambda(\mathbf{x})}$$

以及

$$p_\lambda(y|\mathbf{x}) = \frac{1}{Z_\lambda(\mathbf{x})} \exp\{\sum_{i=1}^{K} \lambda_i f_i(\mathbf{x}, y)\} \tag{7.27}$$

这样我们得到了含有 Lagrange 乘子的模型表达式。下面需要求解外层最优化问题以得到 Lagrange 乘子的值。将外层最优化问题的目标函数写成 λ 的函数

$$\begin{aligned}\Psi(\lambda) =& \mathcal{L}(p_\lambda, \lambda) \\ =& \int \left(\int p_\lambda \log p_\lambda dy \right) \tilde{p}(\mathbf{x})d\mathbf{x} + \sum_{i=1}^{K} \lambda_i (\langle f_i(\mathbf{x}, y) \rangle_{\tilde{p}(y,\mathbf{x})} - \langle f_i(\mathbf{x}, y) \rangle_{\tilde{p}(\mathbf{x})p_\lambda})\end{aligned}$$

为简便起见，我们定义

$$\tau_i = \langle f_i(\mathbf{x}, y) \rangle_{\tilde{p}(y,\mathbf{x})} \tag{7.28}$$

因此有

$$\begin{aligned}\Psi(\lambda) =& \sum_{i=1}^{K} \lambda_i \tau_i + \int \left(\int p_\lambda \log p_\lambda dy \right) \tilde{p}(\mathbf{x})d\mathbf{x} - \sum_{i=1}^{K} \lambda_i \int \tilde{p}(\mathbf{x}) (\int f_i(\mathbf{x}, y) p_\lambda dy) d\mathbf{x} \\ =& \sum_{i=1}^{K} \lambda_i \tau_i + \int \left(\int \left(\log p_\lambda - \sum_{i=1}^{K} \lambda_i f_i(\mathbf{x}, y) \right) p_\lambda dy \right) \tilde{p}(\mathbf{x})d\mathbf{x}\end{aligned}$$

由于

$$\log p_\lambda(y|\mathbf{x}) = \sum_{i=1}^{K} \lambda_i f_i(\mathbf{x}, y) - \log Z_\lambda(\mathbf{x})$$

有

$$\begin{aligned}\Psi(\lambda) =& \sum_{i=1}^{K} \lambda_i \tau_i - \int \left(\int \log Z_\lambda(\mathbf{x}) p_\lambda dy \right) \tilde{p}(\mathbf{x})d\mathbf{x} \\ =& \sum_{i=1}^{K} \lambda_i \tau_i - \int p_\lambda dy \int \log Z_\lambda(\mathbf{x}) \tilde{p}(\mathbf{x})d\mathbf{x} \\ =& \sum_{i=1}^{K} \lambda_i \tau_i - \int \log Z_\lambda(\mathbf{x}) \tilde{p}(\mathbf{x})d\mathbf{x}\end{aligned}$$

代入 $Z_\lambda(\mathbf{x})$ 的定义，我们有

$$\Psi(\lambda)=\sum_{i=1}^{K}\lambda_i\tau_i-\int\log\left(\int\exp\{\sum_{i=1}^{K}\lambda_i f_i(\mathbf{x},y)\}dy\right)\tilde{p}(\mathbf{x})d\mathbf{x}$$

由于 $\mathbf{x},y$ 均为离散取值，可以将积分写成求和形式，有

$$\Psi(\lambda)=\sum_{i=1}^{K}\lambda_i\tau_i-\sum_{k=1}^{N}\frac{1}{N}\log\left(\sum_{y}\exp\{\sum_{i=1}^{K}\lambda_i f_i(\mathbf{x}^{(k)},y)\}\right)\tag{7.29}$$

对 λ_i 求梯度，有

$$\frac{\partial}{\partial\lambda_i}\Psi(\lambda)=\tau_i-\sum_{k=1}^{N}\frac{1}{N}\frac{\sum_y f_i(\mathbf{x}^{(k)},y)\exp\{\sum_{i=1}^{K}\lambda_i f_i(\mathbf{x}^{(k)},y)\}}{\sum_y\exp\{\sum_{i=1}^{K}\lambda_i f_i(\mathbf{x}^{(k)},y)\}}$$

代入 τ_i，可以进行梯度上升迭代求解，即

$$\lambda_i(t+1)=\lambda_i(t)+\alpha\sum_{k=1}^{N}\frac{1}{N}\left(f_i(\mathbf{x}^{(k)},y^{(k)})-\sum_y f_i(\mathbf{x}^{(k)},y)p_{\lambda(t)}(y|\mathbf{x}^{(k)})\right)\tag{7.30}$$

7.4.2 改进迭代尺度法 IIS

IIS 方法从另一个角度考虑优化问题。此前介绍 E-M 算法时，我们指出了 E-M 算法的本质是通过优化似然的一个下界来优化似然函数。此处我们可以使用类似的思路：优化 $\Psi(\lambda)$ 的一个变化量下界从而优化 Ψ。

算法	**最大熵-改进迭代尺度法**
算法简述	对于分类问题，求解条件熵最大的判别模型，通过优化目标变化量下界简化计算
已知	样本 $X\in R^{n\times N}$，标签 $Y=[\mathbf{y^{(1)}},\mathbf{y^{(2)}},...,\mathbf{y^{(N)}}]\in R^{c\times N}$， 特征函数 $f_1(\mathbf{x},y),...,f_K(\mathbf{x},y)$
求	判别模型 $p(y\|\mathbf{x},\lambda)$
解类型	迭代解
算法步骤	1. 对所有样本和所有标签取值计算补特征函数 $C=\max_{k,y}\sum_{i=1}^{K}f_i(\mathbf{x}^{(k)},y)$ $f_{K+1}(\mathbf{x}^{(k)},y)=C-\sum_{i=1}^{K}f_i(\mathbf{x}^{(k)},y)$ 2. 随机初始化参数 $\lambda(0)\in R^{K+1}$ 3. 对每一样本计算模型预测的标签分布 $Z_\lambda(\mathbf{x})=\int\exp\{\sum_{i=1}^{K}\lambda_i f_i(\mathbf{x},y)\}dy$ $p_\lambda(y\|\mathbf{x})=\frac{1}{Z_\lambda(\mathbf{x})}\exp\{\sum_{i=1}^{K}\lambda_i f_i(\mathbf{x},y)\}$ 4. 按下式进行参数迭代 $\delta_i(t)=\frac{1}{C}\log\frac{\sum_{k=1}^{N}f_i(\mathbf{x}^{(k)},y^{(k)})}{\sum_{j=1}^{N}\sum_y p_{\lambda(t)}(y\|\mathbf{x}^{(j)})f_i(\mathbf{x}^{(j)},y)}$ $\lambda_i(t+1)=\lambda_i(t)+\delta_i(t)$ 5. $t\leftarrow t+1$, 若参数收敛，停止迭代，否则回到 3.

具体来说，假设 λ 的一个小变化量为 δ，我们有

$$\Psi(\lambda+\delta)-\Psi(\lambda)=\sum_{i=1}^{K}\delta_i\tau_i-\sum_{k=1}^{N}\frac{1}{N}(\log\frac{Z_{\lambda+\delta}}{Z_\lambda})$$

由于 $-\log a\geq 1-a,\forall a$，我们可以写出变化量的一个下界

$$\Psi(\lambda+\delta)-\Psi(\lambda)\geq A(\delta,\lambda)=\sum_{i=1}^{K}\delta_i\tau_i+1-\sum_{k=1}^{N}\frac{1}{N}\frac{Z_{\lambda+\delta}}{Z_\lambda}$$

由于

$$\begin{aligned}\frac{Z_{\lambda+\delta}}{Z_\lambda}=&\frac{\int\exp\{\sum_{i=1}^{K}(\lambda_i+\delta_i)f_i(\mathbf{x},y)\}dy}{Z_\lambda}\\=&\int\frac{1}{Z_\lambda}\exp\{\sum_{i=1}^{K}\lambda_i f_i(\mathbf{x},y)\}\exp\{\sum_{i=1}^{K}\delta_i f_i(\mathbf{x},y)\}dy\\=&\int p_\lambda(y|\mathbf{x})\exp\{\sum_{i=1}^{K}\delta_i f_i(\mathbf{x},y)\}dy\end{aligned}$$

因此有

$$A(\delta,\lambda)=\sum_{i=1}^{K}\delta_i\tau_i+1-\sum_{k=1}^{N}\frac{1}{N}\int p_\lambda(y|\mathbf{x})\exp\{\sum_{i=1}^{K}\delta_i f_i(\mathbf{x},y)\}dy$$

这一下界的提升仍然较为困难，考虑对其继续寻找下界。定义特征和函数

$$f_S(\mathbf{x},y)=\sum_{i=1}^{K}f_i(\mathbf{x},y)\tag{7.31}$$

一般对于不同的 $\mathbf{x},y$ 取值，$f_S(\mathbf{x},y)$ 取值不同。定义一个新的特征

$$f_{K+1}(\mathbf{x},y)=C-f_S(\mathbf{x},y)\tag{7.32}$$

其中

$$C=\max_{\mathbf{x},y}=f_S(\mathbf{x},y)$$

这样有

$$\sum_{i=1}^{K+1}f_i(\mathbf{x},y)=C,\forall\mathbf{x},y$$

扩充原来的模型为

$$\begin{aligned}p_\lambda(y|\mathbf{x})=&\frac{1}{Z_\lambda(\mathbf{x})}\exp\{\sum_{i=1}^{K+1}\lambda_i f_i(\mathbf{x},y)\}\\Z_\lambda(\mathbf{x})=&\int\exp\{\sum_{i=1}^{K+1}\lambda_i f_i(\mathbf{x},y)\}dy\\\Psi(\lambda)=&\sum_{i=1}^{K+1}\lambda_i\tau_i-\int\log Z_\lambda(\mathbf{x})\tilde{p}(\mathbf{x})d\mathbf{x}\end{aligned}\tag{7.33}$$

得到

$$A(\delta,\lambda)=\sum_{i=1}^{K+1}\delta_i\tau_i+1-\int\int p_\lambda(y|\mathbf{x})\exp\{\sum_{i=1}^{K+1}\delta_i f_i(\mathbf{x},y)\}dyd\mathbf{x}$$

因此我们可以对指数-求和部分找到一个界

$$\begin{aligned}\exp\{\sum_{i=1}^{K+1}\delta_i f_i(\mathbf{x},y)\}=&\exp\{\sum_{i=1}^{K+1}C\delta_i\frac{f_i(\mathbf{x},y)}{C}\}\\ \le&\sum_{i=1}^{K+1}\frac{f_i(\mathbf{x},y)}{C}\exp\{C\delta_i\}\end{aligned}$$

这里我们再次用到了 Jensen 不等式（见2.3.2节）。因此，我们对于 $\Psi(\lambda)$ 的变化量有进一步的下界

$$\Psi(\lambda+\delta)-\Psi(\lambda)\ge A(\delta,\lambda)\ge B(\delta,\lambda)$$

其中

$$\begin{aligned}B(\delta,\lambda)=&\sum_{i=1}^{K+1}\delta_i\tau_i+1-\int\int p(\mathbf{x})p_\lambda(y|\mathbf{x})\sum_{i=1}^{K+1}\frac{f_i(\mathbf{x},y)}{C}\exp\{C\delta_i\}dyd\mathbf{x}\\ =&1+\sum_{i=1}^{K+1}\left(\delta_i\tau_i-\int\int p(\mathbf{x})p_\lambda(y|\mathbf{x})\frac{f_i(\mathbf{x},y)}{C}\exp\{C\delta_i\}dyd\mathbf{x}\right)\\ =&1+\sum_{i=1}^{K+1}\left(\delta_i\tau_i-\int\int\tilde{p}(\mathbf{x})p_\lambda(y|\mathbf{x})\frac{f_i(\mathbf{x},y)}{C}dyd\mathbf{x}\exp\{C\delta_i\}\right)\\ =&1+\sum_{i=1}^{K+1}\left(\delta_i\tau_i-\frac{1}{C}\exp\{C\delta_i\}\left\langle f_i(\mathbf{x},y)\right\rangle_{\tilde{p}(\mathbf{x})p_\lambda(y|\mathbf{x})}\right)\end{aligned}$$

$B(\delta,\lambda)$ 是 δ 的凸函数，存在全局最大值，最大值处偏导为 0

$$\frac{\partial B}{\partial\delta_i}=\tau_i-\exp\{C\delta_i\}\left\langle f_i(\mathbf{x},y)\right\rangle_{\tilde{p}(\mathbf{x})p_\lambda(y|\mathbf{x})}=0$$

即

$$\delta_i=\frac{1}{C}\log\frac{\langle f_i(\mathbf{x},y)\rangle_{\tilde{p}(y,\mathbf{x})}}{\langle f_i(\mathbf{x},y)\rangle_{\tilde{p}(\mathbf{x})p_\lambda(y|\mathbf{x})}}$$

于是我们可以得到更新模型参数的迭代式

$$\delta_i(t)=\frac{1}{C}\log\frac{\sum_{k=1}^{N}f_i(\mathbf{x}^{(k)},y^{(k)})}{\sum_{j=1}^{N}\sum_y p_{\lambda(t)}(y|\mathbf{x}^{(j)})f_i(\mathbf{x}^{(j)},y)}\tag{7.34}$$

$$\lambda_i(t+1)=\lambda_i(t)+\delta_i(t)\tag{7.35}$$

7.4.3 最大熵模型与 Softmax 回归

展开来写最大熵模型给出的判别概率

$$p_\lambda(y|\mathbf{x}) = \frac{\exp\{\sum_{i=1}^{K} \lambda_i f_i(\mathbf{x}, y)\}}{\sum_{y'} \exp\{\sum_{i=1}^{K} \lambda_i f_i(\mathbf{x}, y')\}}$$

假设 y 的取值为 $\{1, 2, ..., C\}$，样本 $\mathbf{x}$ 维度为 n。有矩阵 $W \in R^{C \times n}$。特征函数个数 $K = n$，定义为

$$f_i(\mathbf{x}, y = j) = w_{j,i} x_i$$

取 $\lambda_1 = ... = \lambda_n = 1$，则有

$$p_\lambda(y = j|\mathbf{x}) = \frac{\exp\{\mathbf{w}_j^T \mathbf{x}\}}{\sum_{k=1}^{C} \exp\{\mathbf{w}_j^T \mathbf{x}\}}$$

这就是式3.7给出的 Softmax 回归模型。因此我们可以看出，最大熵模型和 Softmax 回归有着紧密的联系。同理也可以推导最大熵模型和 Logistic 回归之间的联系。

8 高斯概率模型

本章中我们介绍一些和高斯分布密切相关的概率机器学习模型。它们通过高斯分布假设，可以解决回归、分类、聚类、降维等不同任务。

8.1 高斯判别分析 GDA

算法	**高斯判别分析**
算法简述	分类问题中，假设每类均服从高斯分布，求出最大似然分布参数
已知	样本 $X \in R^{n\times N}$（不扩充）、标签 $y^{(i)} \in \{1,2,...,K\}$
求	类别先验 $\pi_1,\pi_2,...,\pi_k \in [0,1], \sum_{j=1}^{n}\pi_j = 1$、聚类均值 $\mu_1,\mu_2,...,\mu_k \in R^n$、聚类方差 Σ，使得似然函数最大化 $\max p(X,Y\|\pi,\mu,\Sigma)$
解类型	闭式解
闭式解	$\mu_j = \frac{\sum_{i=1}^{N} I(y^{(i)}=j)\mathbf{x}^{(i)}}{\sum_{i=1}^{N} I(y^{(i)}=j)}$,$\pi_j = \frac{\sum_{i=1}^{N} I(y^{(i)}=j)}{N}$, $\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)}-\mu_j)(\mathbf{x}^{(i)}-\mu_j)^T$

GDA 假设每类样本均服从样本方差相同的高斯分布

$$p(\mathbf{x}|y) = \prod_{j=1}^{K} N(\mathbf{x}|\mu_j,\Sigma)^{I(y=j)}$$

同时，假设有类别先验

$$p(y) = \prod_{j=1}^{K} \pi_j^{I(y=j)}$$

其中 $I(\cdot)$ 是判别函数，π 是分布参数，满足

$$\sum_{j=1}^{K}\pi_j = 1$$

则有联合分布

$$p(y,\mathbf{x}|\mu,\Sigma,\pi) = p(\mathbf{x}|y,\mu,\Sigma)p(y|\pi)$$

对数似然函数

$$\begin{aligned}
\log L(\mu,\Sigma,\pi) &= \log\prod_{i=1}^{N} p(y^{(i)},\mathbf{x}^{(i)}|\mu,\Sigma,\pi) \\
&= \sum_{i=1}^{N}(\log p(\mathbf{x}^{(i)}|y^{(i)},\mu,\Sigma) + \log p(y^{(i)}|\pi)) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{K} I(y^{(i)}=j)(\log\pi_j - \frac{1}{2}(k\log 2\pi + \log|\Sigma|) - \frac{1}{2}(\mathbf{x}^{(i)}-\mu_j)^T\Sigma^{-1}(\mathbf{x}^{(i)}-\mu_j))
\end{aligned}$$

对各参数求偏导，首先是均值

$$\frac{\partial \log L}{\partial \mu_j} = \sum_{i=1}^{N} I(y^{(i)} = j)\Sigma^{-1}(\mathbf{x}^{(i)} - \mu_j) = 0$$

$$\sum_{i=1}^{N} I(y^{(i)} = j)\mu_j = \sum_{i=1}^{N} I(y^{(i)} = j)\mathbf{x}^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{N} I(y^{(i)} = j)\mathbf{x}^{(i)}}{\sum_{i=1}^{N} I(y^{(i)} = j)}$$

均值的最优估计实际上就是该类样本的样本均值。
其次是方差

$$\frac{\partial \log L}{\partial \Sigma}(-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{K} I(y^{(i)} = j)(\log|\Sigma| + (\mathbf{x}^{(i)} - \mu_j)^T\Sigma^{-1}(\mathbf{x}^{(i)} - \mu_j))) = 0$$

$$\sum_{i=1}^{N} \frac{\partial \log L}{\partial \Sigma}\log|\Sigma| = \sum_{i=1}^{N} \frac{\partial \log L}{\partial \Sigma}(\mathbf{x}^{(i)} - \mu_j)^T\Sigma^{-1}(\mathbf{x}^{(i)} - \mu_j)$$

$$N\frac{1}{|\Sigma|}|\Sigma|\Sigma^{-1} = \sum_{i=1}^{N} \Sigma^{-1}(\mathbf{x}^{(i)} - \mu_j)(\mathbf{x}^{(i)} - \mu_j)^T\Sigma^{-1}$$

$$N\Sigma = \sum_{i=1}^{N}(\mathbf{x}^{(i)} - \mu_j)(\mathbf{x}^{(i)} - \mu_j)^T$$

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \mu_j)(\mathbf{x}^{(i)} - \mu_j)^T$$

方差的最大似然估计其实也就是样本方差。
最后，对于 π，这时一个约束优化问题，有

$$\begin{aligned} \min_{\pi} \quad & L(\pi) = \sum_{i=1}^{N}\sum_{j=1}^{K} I(y^{(i)} = j)\log \pi_j \\ \text{s.t.} \quad & \sum_{j=1}^{K} \pi_j = 1 \end{aligned}$$

要构造 Lagrange 函数

$$\begin{aligned} &\mathcal{L}(\pi) = \log L + \lambda(\sum_{k=1}^{K} \pi_k - 1) \\ =&\sum_{i=1}^{N}\sum_{j=1}^{K} I(y^{(i)} = j)\log \pi_j + \lambda(\sum_{k=1}^{K} \pi_k - 1) + C \end{aligned}$$

求偏导

$$\frac{\partial \mathcal{L}}{\partial \pi_j} = \sum_{i=1}^{N} I(y^{(i)} = j)\frac{1}{\pi_j} + \lambda = 0$$
$$\pi_j = -\frac{\sum_{i=1}^{N} I(y^{(i)} = j)}{\lambda}$$

由于

$$\sum_{j=1}^{K} \pi_j = \sum_{j=1}^{K} -\frac{\sum_{i=1}^{N} I(y^{(i)} = j)}{\lambda} = 1$$

有

$$\lambda = -\sum_{j=1}^{K}\sum_{i=1}^{N} I(y^{(i)} = j) = -N$$

因此

$$\pi_j = \frac{\sum_{i=1}^{N} I(y^{(i)} = j)}{N}$$

所以我们有最大似然解

$$\begin{aligned}
\mu_j &= \frac{\sum_{i=1}^{N} I(y^{(i)} = j)\mathbf{x}^{(i)}}{\sum_{i=1}^{N} I(y^{(i)} = j)} \\
\Sigma &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \mu_j)(\mathbf{x}^{(i)} - \mu_j)^T \\
\pi_j &= \frac{\sum_{i=1}^{N} I(y^{(i)} = j)}{N}
\end{aligned} \tag{8.1}$$

8.2 高斯过程

想象一个由无穷个随机变量组成的无穷维高斯分布。其中随机变量 $\mathbf{t}$ 可以由连续索引值 $\mathbf{x} \in R^n$ 索引得到 $\mathbf{t}_\mathbf{x}$。假定该高斯分布的均值对任意维度为 0，任意两个维度 $\mathbf{x}_1, \mathbf{x}_2$ 间的协方差由核函数 $\kappa(\cdot, \cdot)$ 计算，即

$$\mathrm{Cov}(\mathbf{t}_{\mathbf{x}_1}, \mathbf{t}_{\mathbf{x}_2}) = \kappa(\mathbf{x}_1, \mathbf{x}_2) \tag{8.2}$$

这样的一列 $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t$ 组成的随机序列 $\{\mathbf{x}_t\}$ 被称为高斯过程。

简单起见，考虑 $n = 1$，则有 $\mathbf{t}$ 的索引范围是整个实数集。根据无穷维高斯分布的假设，任取实数集中的两个实数 $r_1, r_2 \in R$。都可以找到两个随机变量 $\mathbf{t}_{r_1}$，$\mathbf{t}_{r_2}$ 满足高斯分布

$$\mathbf{t}_{r_1} \sim N(\mathbf{t}_{r_1}; 0, \kappa(r_1, r_1)),\ \mathbf{t}_{r_2} \sim N(\mathbf{t}_{r_2}; 0, \kappa(r_2, r_2))$$

且两个变量的联合分布也满足高斯分布

$$\mathbf{t}_{r_1}, \mathbf{t}_{r_2} \sim N(\mathbf{t}_{r_1}, \mathbf{t}_{r_1}; 0, K)$$

其中

$$K = \begin{bmatrix} \kappa(r_1, r_1) & \kappa(r_1, r_2) \\ \kappa(r_2, r_1) & \kappa(r_2, r_2) \end{bmatrix} \tag{8.3}$$

虽然我们假定了一个无限维度的高斯分布，但实际使用中我们只会用到有限个维度，并不会出现无法计算的问题。

8.2.1 高斯过程回归 GPR

算法	**高斯过程回归**
算法简述	回归问题中，假设样本构成高斯过程，标签是对样本索引维度的随机变量的观测，求出新样本下新标签的后验条件分布
已知	样本 $X \in R^{n \times N}$（不扩充）、标签 $y^{(i)} \in R$
求	新标签的后验条件分布 $p(y|\mathbf{x}, X, Y)$
解类型	闭式解
闭式解	$p(y|\mathbf{x}, X, Y) = N(y; \mathbf{k}^T(K_N + \alpha I_N)^{-1}\mathbf{y}_N, k - \mathbf{k}^T(K_N + \alpha I_N)^{-1}\mathbf{k})$

高斯过程可以用于对回归问题进行建模。假设拟合一维变量 $y(i) \in R$，那么将样本集 $X \in R^{n \times N}$ 构成高斯过程，标签 $y(i)$ 可以认为是对随机变量 $\mathbf{t}_{\mathbf{x}^{(i)}}$ 的一次抽样。假设抽样时有随机噪声 $\epsilon \sim N(0, \alpha)$，即

$$y^{(i)} = \mathbf{t}_{\mathbf{x}^{(i)}} + \epsilon \tag{8.4}$$

根据高斯过程假设，N 个样本索引的随机变量构成联合分布

$$p(t_{\mathbf{x}^{(1)}}, t_{\mathbf{x}^{(2)}}, ..., t_{\mathbf{x}^{(N)}}) = N(\mathbf{t}_N; \mathbf{0}, K_N)$$

这里 K_N 是这 N 个随机变量之间的协方差矩阵，由索引通过核函数计算得到，即

$$\mathrm{Cov}(t_{\mathbf{x}^{(i)}}, t_{\mathbf{x}^{(j)}}) = (K_N)_{ij} = \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

考虑噪声影响，由于两个高斯分布的和仍然是高斯分布，可以写出 $y^{(i)}$ 的分布

$$p(y^{(i)}) = N(y^{(i)}; 0, \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}) + \alpha)$$

那么，回归问题就变成了求有限个维度的观测条件下的标签后验分布问题。对于新的样本 $\mathbf{x}$，它索引的随机变量和现有随机变量间的协方差向量为

$$\mathbf{k} = [\kappa(\mathbf{x}, \mathbf{x}^{(1)}), \kappa(\mathbf{x}, \mathbf{x}^{(2)}), ..., \kappa(\mathbf{x}, \mathbf{x}^{(N)})]^T \tag{8.5}$$

其自身方差为

$$k = \mathrm{Cov}(t_{\mathbf{x}}, t_{\mathbf{x}}) = \kappa(\mathbf{x}, \mathbf{x}) \tag{8.6}$$

定义矩阵

$$K_{N+1} = \begin{bmatrix} K_N & \mathbf{k} \\ \mathbf{k}^T & k \end{bmatrix} \tag{8.7}$$

则加上新样本索引的随机变量后，联合分布仍为高斯分布

$$p(t_{\mathbf{x}^{(1)}}, t_{\mathbf{x}^{(2)}}, ..., t_{\mathbf{x}^{(N)}}, t_{\mathbf{x}}) = N(\mathbf{t}_{N+1}; \mathbf{0}, K_{N+1})$$

再考虑噪声的影响。记 $\mathbf{y}_N = [y^{(1)}, y^{(2)}, ..., y^{(N)}]^T, \mathbf{y}_{N+1} = [\mathbf{y}_N^T, y]^T$。为简便起见，记

$$\Sigma_N = K_N + \alpha I_N, \Sigma_{N+1} = K_{N+1} + \alpha I_{N+1}$$

由于不同采样间的噪声是独立同分布的，因此有

$$p(y^{(1)}, y^{(2)}, ..., y^{(N)}, y) = N(\mathbf{y}_{N+1}; \mathbf{0}, \Sigma_{N+1})$$

因此，可以计算条件概率

$$\begin{aligned}
&p(y|\mathbf{x}, X, Y) = p(t_{\mathbf{x}} = y | t_{\mathbf{x}^{(1)}} = y^{(1)}, t_{\mathbf{x}^{(2)}} = y^{(2)}, ..., t_{\mathbf{x}^{(N)}} = y^{(N)}) \\
=&\frac{p(t_{\mathbf{x}^{(1)}} = y^{(1)}, t_{\mathbf{x}^{(2)}} = y^{(2)}, ..., t_{\mathbf{x}^{(N)}} = y^{(N)}, t_{\mathbf{x}} = y)}{p(t_{\mathbf{x}^{(1)}} = y^{(1)}, t_{\mathbf{x}^{(2)}} = y^{(2)}, ..., t_{\mathbf{x}^{(N)}} = y^{(N)})} \\
=&\frac{\frac{1}{\sqrt{(2\pi)^{N+1}|\Sigma_{N+1}|}} \exp\{\frac{1}{2}\mathbf{y}_{N+1}^T \Sigma_{N+1}^{-1} \mathbf{y}_{N+1}\}}{\frac{1}{\sqrt{(2\pi)^N |\Sigma_N|}} \exp\left\{\frac{1}{2}\mathbf{y}_N^T \Sigma_N^{-1} \mathbf{y}_N\right\}} \\
=&\sqrt{\frac{|\Sigma_N|}{2\pi|\Sigma_{N+1}|}} \exp\left\{\frac{1}{2}(\mathbf{y}_{N+1}^T \Sigma_{N+1}^{-1} \mathbf{y}_{N+1} - \mathbf{y}_N^T \Sigma_N^{-1} \mathbf{y}_N)\right\}
\end{aligned}$$

根据分块矩阵求逆公式2.4，有

$$\begin{aligned}
&\mathbf{y}_{N+1}^T (\Sigma_{N+1})^{-1} \mathbf{y}_{N+1} = \begin{bmatrix}\mathbf{y}_N^T & y\end{bmatrix} \begin{bmatrix}\Sigma_N & \mathbf{k} \\ \mathbf{k}^T & k+\alpha\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{y}_N \\ y\end{bmatrix} \\
=&\begin{bmatrix}\mathbf{y}_N^T & y\end{bmatrix} \begin{bmatrix}\Sigma_N^{-1} + \Sigma_N^{-1}\mathbf{k}\gamma^{-1}\mathbf{k}^T\Sigma_N^{-1} & -\Sigma_N^{-1}\mathbf{k}\gamma^{-1} \\ -\gamma^{-1}\mathbf{k}^T\Sigma_N^{-1} & \gamma^{-1}\end{bmatrix} \begin{bmatrix}\mathbf{y}_N \\ y\end{bmatrix} \\
=&\ \mathbf{y}_N^T\Sigma_N^{-1}\mathbf{y}_N + \mathbf{y}_N^T(\Sigma_N^{-1}\mathbf{k}\gamma^{-1}\mathbf{k}^T\Sigma_N^{-1})\mathbf{y}_N \\
&- \mathbf{y}_N^T\Sigma_N^{-1}\mathbf{k}\gamma^{-1}y - y\gamma^{-1}\mathbf{k}^T\Sigma_N^{-1}\mathbf{y}_N + \gamma^{-1}y^2 \\
=&\ \mathbf{y}_N^T\Sigma_N^{-1}\mathbf{y}_N + \gamma^{-1}(y - \mathbf{k}^T\Sigma_N^{-1}\mathbf{y}_N)^2
\end{aligned}$$

其中

$$\gamma = k + \alpha - \mathbf{k}^T\Sigma_N^{-1}\mathbf{k} \tag{8.8}$$

又由分块矩阵行列式性质2.3，有

$$|\Sigma_{N+1}| = |K_N + \alpha I_N||k + \alpha - \mathbf{k}^T(K_N + \alpha I_N)^{-1}\mathbf{k}| = \gamma|\Sigma_N|$$

因此条件概率

$$p(y|\mathbf{x}, X, Y) = \sqrt{\frac{1}{2\pi\gamma}} \exp\{\frac{1}{2\gamma}(y - \mathbf{k}^T\Sigma_N^{-1}\mathbf{y}_N)^2\}$$

即

$$y|\mathbf{x}, X, Y \sim N(y; \mu_y, \Sigma_y) \tag{8.9}$$

其中

$$\begin{aligned}\mu_y =& \mathbf{k}^T(K_N + \alpha I_N)^{-1}\mathbf{y}_N \\ \Sigma_y =& \alpha + k - \mathbf{k}^T(K_N + \alpha I_N)^{-1}\mathbf{k}\end{aligned} \tag{8.10}$$

如果要对样本给出预测的话，使用分布均值即可。上式也可以使用结论2.37直接得到。

8.2.2 GPR 与线性回归

如果我们取核函数

$$\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = {\mathbf{x}^{(i)}}^T\mathbf{x}^{(j)}$$

那么有

$$K_N = X^TX, \mathbf{k} = X^T\mathbf{x}, k = \mathbf{x}^T\mathbf{x}$$

因此后验分布的均值为

$$\begin{aligned}\mu_y =& \mathbf{k}^T(\Sigma_{N+1})^{-1}\mathbf{y}_N \\ =& \mathbf{x}^TX(X^TX + \alpha I)^{-1}\mathbf{y}_N\end{aligned}$$

方差为

$$\begin{aligned}\Sigma_y =& k + \alpha - \mathbf{k}^T(\Sigma_{N+1})^{-1}\mathbf{k} \\ =& \mathbf{x}^T\mathbf{x} + \alpha - \mathbf{x}^TX(X^TX + \alpha I)^{-1}X^T\mathbf{x} \\ =& \alpha + \mathbf{x}^T(I - X(X^TX + \alpha I)^{-1}X^T)\mathbf{x}\end{aligned}$$

即

$$\begin{aligned}\mu_y =& \mathbf{x}^TX(\alpha I + X^TX)^{-1}\mathbf{y}_N \\ \Sigma_y =& \alpha + \mathbf{x}^T(I - X(\alpha I + X^TX)^{-1}X^T)\mathbf{x}\end{aligned}$$

这里我们有

$$X(\alpha I + X^TX)^{-1} = (\alpha I + XX^T)^{-1}X$$

因为

$$\begin{aligned}&(\alpha I + XX^T)\left(X(\alpha I + X^TX)^{-1} - (\alpha I + XX^T)^{-1}X\right)(\alpha I + X^TX) \\ &= \alpha X + XX^TX - (\alpha X + XX^TX) = 0\end{aligned}$$

因此我们有

$$\begin{aligned}\mu_y =& \mathbf{x}^TX(X^TX + \alpha I)^{-1}\mathbf{y}_N \\ =& \mathbf{x}^T(\alpha I + XX^T)^{-1}X\mathbf{y}_N\end{aligned}$$

并且

$$\begin{aligned}\Sigma_y =&\alpha + \mathbf{x}^T(I - X(\alpha I + X^T X)^{-1}X^T)\mathbf{x}\\ =&\alpha + \mathbf{x}^T(I - (\alpha I + XX^T)^{-1}XX^T)\mathbf{x}\\ =&\alpha + \mathbf{x}^T(I - (\alpha I + XX^T)^{-1}(XX^T + \alpha I) + \alpha(\alpha I + XX^T)^{-1})\mathbf{x}\\ =&\alpha + \mathbf{x}^T(\alpha(\alpha I + XX^T)^{-1})\mathbf{x}\\ =&\alpha(1 + \mathbf{x}^T(\alpha I + XX^T)^{-1}\mathbf{x})\end{aligned}$$

即

$$\begin{aligned}\mu_y =&\mathbf{x}^T(\alpha I + XX^T)^{-1}X\mathbf{y}_N\\ \Sigma_y =&\alpha(1 + \mathbf{x}^T(\alpha I + XX^T)^{-1}\mathbf{x})\end{aligned} \tag{8.11}$$

对比线性回归预测分布的表达式7.7，若我们取该式中 $\beta = 1$，那么该式中均值和8.11中的均值完全相同，方差只差一个常数。

这说明，高斯过程回归在核函数取向量内积时，退化为线性回归。这个特殊的核函数也被称为线性核函数。从这个方面来说，高斯过程回归拥有比线性回归更强的表达能力。

更进一步来说，在所有使用核技巧的机器学习方法中，将核函数取为线性核函数，都将退化为相应的线性方法。针对本书中已经出现的所有核方法和对应线性方法，列表总结如下

线性方法	**位置**	**核方法**	**位置**
线性回归	3.1.1, 7.2.1	高斯过程回归	8.2.1
支持向量机	3.2.1	核支持向量机	3.2.2
k 近邻回归	3.3.2	核回归	3.3.3
主成分分析	4.1.1	核主成分分析	4.2.1
k 均值聚类	5.1.1	核 k 均值聚类	5.1.3

8.3 高斯混合模型 GMM

高斯混合模型是一种生成式模型，可用于解决聚类问题。高斯混合模型认为，数据集的生成过程是：先从 K 个不同的高斯分布中选择一个分布，再从这一分布中抽样。

$$N(\mathbf{x}|\mu_k, \Sigma_k), k = 1, 2, 3, ..., K$$

因此，样本的来源分布是一个未知的随机变量，即隐变量。将隐变量其记为 0-1 向量 $\mathbf{z} \in \{0, 1\}^K$。$N$ 个隐变量可以组成矩阵 $Z \in \{0, 1\}^{K\times N}$，其中 $z_{ji} = 1$ 表示第 i 个样本是从第 j 个分布中抽样的。则有

$$\sum_{j=1}^{K} z_{ji} = 1, \forall i$$

隐变量 $\mathbf{z}$ 满足多项分布，即

$$p(\mathbf{z}_k = 1) = \pi_k$$

其中 $\pi = [\pi_1, \pi_2, ..., \pi_K]^T$ 是未知参数，满足

$$\sum_{k=1}^{K}\pi_k = 1$$

因此，观测变量和隐变量的联合分布可以写为

$$p(X, Z|\mu, \Sigma, \pi) = \prod_{i=1}^{N}\prod_{j=1}^{K}\pi_j^{z_{ji}} N(\mathbf{x}^{(i)}|\mu_j, \Sigma_j)^{z_{ji}} \tag{8.12}$$

对数似然函数

$$\begin{aligned}
&\log L(\mu, \Sigma, \pi) = \log p(X|\mu, \Sigma, \pi) \\
=&\log \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\mu, \Sigma, \pi) \\
=&\sum_{i=1}^{N} \log \sum_{\mathbf{z}\in\mathbf{Z}} p(\mathbf{x}^{(i)}, \mathbf{z}|\mu, \Sigma, \pi) \\
=&\sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k N(\mathbf{x}^{(i)}|\mu_k, \Sigma_k)
\end{aligned}$$

8.3.1 GMM E-M 算法

对数似然出现了对数内求和的形式，直接优化复杂。因此，考虑使用 E-M 算法（见6.2节）进行求解。

算法	**高斯混合模型-EM 算法**
算法简述	聚类问题中，假设每类均服从高斯分布，迭代更新隐变量期望和分布参数，求出最大似然的分配结果和分布参数
已知	样本 $X \in R^{n\times N}$（不扩充）
求	类别先验 $\pi_1, \pi_2, ..., \pi_k \in [0,1], \sum_{j=1}^{n}\pi_j = 1$、聚类均值 $\mu_1, \mu_2, ..., \mu_k \in R^n$、聚类方差 $\Sigma_1, \Sigma_2, ..., \Sigma_k \in R^{n\times n}$、隐变量 $\mathbf{z}^{(i)} \in \{0,1\}^k, \sum_{j=1}^{k} z_{ij} = 1$，使得似然函数最大化 $\max p(X|\pi, \mu, \Sigma)$
解类型	迭代解
算法步骤	1. 使用 k 均值聚类得到初始聚类结果，初始化均值、方差、类别先验 2. E 步：隐变量期望 $\gamma(z_{ji}) = \dfrac{N(\mathbf{x}^{(i)}\|\mu_j, \Sigma_j)}{\sum_{k=1}^{K} N(\mathbf{x}^{(i)}\|\mu_k, \Sigma_k)}$ 3. M 步：$\mu_j = \dfrac{\sum_{i=1}^{N}\gamma(z_{ji})\mathbf{x}^{(i)}}{\sum_{i=1}^{N}\gamma(z_{ji})}$, $\pi_j = \dfrac{\sum_{i=1}^{N}\gamma(z_{ji})}{\sum_{k=1}^{K}\sum_{i=1}^{N}\gamma(z_{ji})}$, $\Sigma_k = \dfrac{1}{\sum_{i=1}^{N}\gamma(z_{ji})}\sum_{i=1}^{N}\gamma(z_{ji})(\mathbf{x}^{(i)} - \mu_j)(\mathbf{x}^{(i)} - \mu_j)^T$ 4. 若似然函数收敛，停止迭代，计算降维结果。否则返回 2。

首先考虑 M 步。联合分布的对数为

$$\log p(X,Z|\mu,\Sigma,\pi)=\log\prod_{i=1}^{N}\prod_{j=1}^{K}\pi_j^{z_{ji}}N(\mathbf{x}^{(i)}|\mu_j,\Sigma_j)^{z_{ji}}$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{K}z_{ji}(\log\pi_j+\log N(\mathbf{x}^{(i)}|\mu_j,\Sigma_j)^{z_{ji}})$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{K}z_{ji}(\log\pi_j-\frac{1}{2}(n\log 2\pi+\log|\Sigma_k|)-\frac{1}{2}(\mathbf{x}^{(i)}-\mu_j)^T\Sigma_j^{-1}(\mathbf{x}^{(i)}-\mu_j))$$

似然函数的一个下界是该函数对于隐变量求期望。为简便起见，记

$$\gamma(z_{ji})=E_{\mathbf{z}_i\sim p(\mathbf{z}|\mathbf{x}^{(i)},\theta)}[z_{ji}] \tag{8.13}$$

则有

$$LB(\mu,\Sigma,\pi)=E_{Z\sim p(Z|X,\theta)}[\log p(X,Z|\mu,\Sigma,\pi)]$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{K}\gamma(z_{ji})(\log\pi_j-\frac{1}{2}(n\log 2\pi+\log|\Sigma_k|)-\frac{1}{2}(\mathbf{x}^{(i)}-\mu_j)^T\Sigma_j^{-1}(\mathbf{x}^{(i)}-\mu_j))$$

对于不同参数可以分别只考虑和参数有关的项。此处要注意 π 还有额外约束，是约束优化问题。

$$LB(\mu_j)=\sum_{i=1}^{N}-\frac{1}{2}\gamma(z_{ji})(\mathbf{x}^{(i)}-\mu_j)^T\Sigma_j^{-1}(\mathbf{x}^{(i)}-\mu_j)$$

$$LB(\Sigma_j)=\sum_{i=1}^{N}\gamma(z_{ji})(-\frac{1}{2}\log|\Sigma_k|-\frac{1}{2}(\mathbf{x}^{(i)}-\mu_j)^T\Sigma_j^{-1}(\mathbf{x}^{(i)}-\mu_j))$$

$$LB(\pi_j)=\sum_{i=1}^{N}\gamma(z_{ji})\log\pi_j,\ \sum_{k=1}^{K}\pi_k=1$$

在 M 步，隐变量分布使用上一 E 步的估计值，不随参数值改变，因此 $\gamma(z_{ji})$ 可以认为是常数。下面，分别求解各参数。

首先是均值

$$\frac{\partial LB}{\partial\mu_j}=\sum_{i=1}^{N}-\gamma(z_{ji})\Sigma_j^{-1}(\mathbf{x}^{(i)}-\mu_j)=0$$

$$\sum_{i=1}^{N}\gamma(z_{ji})\mu_j=\sum_{i=1}^{N}\gamma(z_{ji})\mathbf{x}^{(i)}$$

$$\mu_j=\frac{\sum_{i=1}^{N}\gamma(z_{ji})\mathbf{x}^{(i)}}{\sum_{i=1}^{N}\gamma(z_{ji})} \tag{8.14}$$

然后是方差

$$\frac{\partial}{\partial\Sigma_j}\left(-\frac{1}{2}\sum_{i=1}^{N}\gamma(z_{ji})\left(\log|\Sigma_j|+(\mathbf{x}^{(i)}-\mu_j)^T\Sigma_j^{-1}(\mathbf{x}^{(i)}-\mu_j)\right)\right)=0$$

$$\sum_{i=1}^{N}\gamma(z_{ji})\frac{\partial LB}{\partial\Sigma_j}\log|\Sigma_j|=\sum_{i=1}^{N}\gamma(z_{ji})\frac{\partial LB}{\partial\Sigma_j}(\mathbf{x}^{(i)}-\mu_j)^T\Sigma_j^{-1}(\mathbf{x}^{(i)}-\mu_j)$$

$$\sum_{i=1}^{N}\gamma(z_{ji})\frac{1}{|\Sigma_j|}|\Sigma_j|\Sigma_k^{-1}=\sum_{i=1}^{N}\gamma(z_{ji})\Sigma_j^{-1}(\mathbf{x}^{(i)}-\mu_j)(\mathbf{x}^{(i)}-\mu_j)^T\Sigma_j^{-1}$$

$$\sum_{i=1}^{N}\gamma(z_{ji})\Sigma_k=\sum_{i=1}^{N}\gamma(z_{ji})(\mathbf{x}^{(i)}-\mu_j)(\mathbf{x}^{(i)}-\mu_j)^T$$

$$\Sigma_k=\frac{1}{\sum_{i=1}^{N}\gamma(z_{ji})}\sum_{i=1}^{N}\gamma(z_{ji})(\mathbf{x}^{(i)}-\mu_j)(\mathbf{x}^{(i)}-\mu_j)^T \tag{8.15}$$

最后，对于 π，这是一个约束优化问题，有

$$\begin{aligned}\min_{\pi}\quad & LB(\pi)=\sum_{i=1}^{N}\sum_{j=1}^{K}\gamma(z_{ji})\log\pi_j\\ \text{s.t.}\quad & \sum_{k=1}^{K}\pi_k=1\end{aligned} \tag{8.16}$$

要构造 Lagrange 函数

$$\begin{aligned}\mathcal{L}(\pi)&=LB(\pi)+\lambda(\sum_{k=1}^{K}\pi_k-1)\\ &=\sum_{i=1}^{N}\sum_{j=1}^{K}\gamma(z_{ji})\log\pi_j+\lambda(\sum_{k=1}^{K}\pi_k-1)\end{aligned}$$

求偏导

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\pi_j}&=\sum_{i=1}^{N}\gamma(z_{ji})\frac{1}{\pi_j}+\lambda=0\\ \pi_j&=-\frac{\sum_{i=1}^{N}\gamma(z_{ji})}{\lambda}\end{aligned}$$

由于

$$\sum_{k=1}^{K}\pi_k=\sum_{k=1}^{K}-\frac{\sum_{i=1}^{N}\gamma(z_{ji})}{\lambda}=1$$

有

$$\lambda=-\sum_{k=1}^{K}\sum_{i=1}^{N}\gamma(z_{ji})$$

因此

$$\pi_j=\frac{\sum_{i=1}^{N}\gamma(z_{ji})}{\sum_{k=1}^{K}\sum_{i=1}^{N}\gamma(z_{ji})} \tag{8.17}$$

由此，我们得到 M 步的迭代公式

$$
\begin{aligned}
\mu_j &= \frac{\sum_{i=1}^{N}\gamma(z_{ji})\mathbf{x}^{(i)}}{\sum_{i=1}^{N}\gamma(z_{ji})} \\
\Sigma_k &= \frac{1}{\sum_{i=1}^{N}\gamma(z_{ji})}\sum_{i=1}^{N}\gamma(z_{ji})(\mathbf{x}^{(i)}-\mu_j)(\mathbf{x}^{(i)}-\mu_j)^T \\
\pi_j &= \frac{\sum_{i=1}^{N}\gamma(z_{ji})}{\sum_{k=1}^{K}\sum_{i=1}^{N}\gamma(z_{ji})}
\end{aligned}
\tag{8.18}
$$

E 步，计算隐变量条件期望，即计算第 i 个样本属于第 j 个高斯分布的概率。我们直接使用贝叶斯公式

$$
\begin{aligned}
\gamma(z_{ji}) =& E_{\mathbf{z}_i\sim p(\mathbf{z}|\mathbf{x}^{(i)},\theta)}[z_{ji}] \\
=& p(z_{ji}=1|\mathbf{x}^{(i)},\theta) \\
=& \frac{p(\mathbf{x}^{(i)},z_{ji}=1|\theta)}{\sum_{k=1}^{K}p(\mathbf{x}^{(i)},z_{ki}=1|\theta)} \\
=& \frac{N(\mathbf{x}^{(i)}|\mu_j,\Sigma_j)}{\sum_{k=1}^{K}N(\mathbf{x}^{(i)}|\mu_k,\Sigma_k)}
\end{aligned}
\tag{8.19}
$$

迭代算法的停止判定条件是，对数似然函数 $\log L(\mu,\Sigma,\pi)$ 收敛。

8.4 因子分析 FA

因子分析是一种用于线性降维的概率模型，是一种生成式模型。

假设有数据集 X，因子分析认为 $\mathbf{x}\in R^n$ 是由低维空间服从标准正态分布的 d 个独立维度经过线性变换生成的，同时加上高维的一些噪声。

具体来说，设随机向量 $\mathbf{z}\in R^d\sim N(0,I_d)$，矩阵 $W\in R^{n\times d}$，噪声 $\epsilon\sim N(0,\Psi)$，向量 $\mu\in R^n$，则 $\mathbf{x}$ 满足

$$
\mathbf{x}=\mu+W\mathbf{z}+\epsilon \tag{8.20}
$$

因此 $\mathbf{x}$ 的均值

$$
\mu_x = E[\mu]+E[W\mathbf{z}]+E[\epsilon]=\mu
$$

方差

$$
\begin{aligned}
\Sigma_x =& E[(\mathbf{x}-\mu_x)(\mathbf{x}-\mu_x)^T] \\
=& E[(W\mathbf{z}+\epsilon)(W\mathbf{z}+\epsilon)^T] \\
=& E[W\mathbf{z}^T\mathbf{z}W^T]+E[\epsilon W\mathbf{z}]+E[W\mathbf{z}\epsilon^T W^T]+E[\epsilon^T\epsilon] \\
=& WE[\mathbf{z}^T\mathbf{z}]W^T+E[\epsilon^T\epsilon]=WW^T+\Psi
\end{aligned}
$$

高斯分布的线性组合仍然是高斯分布，因此有边缘分布

$$
\mathbf{x}\sim N(\mu,WW^T+\Psi) \tag{8.21}
$$

这就是数据的生成模型。下面，需要估计三个参数 μ,W,Ψ。

8.4.1 最大似然

写出负对数似然（简单起见只考虑一个样本 $\mathbf{x}$）

$$\begin{aligned}&\min_{\mu,W,\Psi} L(\mu,W,\Psi) = -\log p(\mathbf{x}|\mu,W,\Psi)\\ =&\frac{1}{2}(n\log(2\pi)+\log|WW^T+\Psi|)+\frac{1}{2}(\mathbf{x}-\mu)^T(WW^T+\Psi)^{-1}(\mathbf{x}-\mu)\end{aligned}$$

对 μ 求偏导

$$\frac{\partial L}{\partial \mu} = (WW^T+\Psi)^{-1}(\mu-\mathbf{x})$$

令偏导为 0，考虑多个样本梯度叠加，有

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)} \tag{8.22}$$

对 W,Ψ 求偏导前，先考虑

$$\begin{aligned}&\frac{\partial L}{\partial \Sigma_x} = \frac{1}{2}\frac{\partial}{\partial \Sigma_x}\left(\log|\Sigma_x|+(\mathbf{x}-\mu)^T\Sigma_x^{-1}(\mathbf{x}-\mu)\right)\\ =&\frac{1}{2}\left(\frac{\partial}{\partial \Sigma_x}\log|\Sigma_x|+\frac{\partial}{\partial \Sigma_x}(\mathbf{x}-\mu)^T\Sigma_x^{-1}(\mathbf{x}-\mu)\right)\\ =&\frac{1}{2}\left(\frac{1}{|\Sigma_x|}|\Sigma_x|\Sigma_x^{-1}+\Sigma_x^{-1}(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T\Sigma_x^{-1}\right)\\ =&\frac{1}{2}\Sigma_x^{-1}(\Sigma_x-(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T)\Sigma_x^{-1}\end{aligned}$$

由于

$$\Sigma_x = \Psi + WW^T \tag{8.23}$$

可知对数似然对 Ψ 和 W 的依赖关系很复杂，无法解析求解。

8.4.2 因子分析 E-M 算法

由于因子分析也是一个含隐变量的生成模型，可以考虑使用 E-M 算法求解参数。考虑使用条件函数版本的优化目标

$$CF(W,\Psi) = \langle \log p(\mathbf{x}|\mathbf{z},W,\Psi)\rangle_{p(\mathbf{x},\mathbf{z}|W_{old},\Psi_{old})}$$

我们有 $\mathbf{x}$ 在 $\mathbf{z}$ 下的条件分布

$$\mathbf{x}|\mathbf{z} \sim N(\mathbf{x};\mu+W\mathbf{z},\Psi)$$

因此

$$\log p(\mathbf{x}|\mathbf{z}) = -\frac{1}{2}(n\log(2\pi)+\log|\Psi|)-\frac{1}{2}(\mathbf{x}-\mu-W\mathbf{z})^T\Psi^{-1}(\mathbf{x}-\mu-W\mathbf{z})$$

接下来，我们需要对隐变量对于后验分布求期望。隐变量和观测变量的联合分布是高斯分布，有

算法	**因子分析-EM 算法**
算法简述	对于降维问题，假设数据是由降维后的 d 个独立维度经线性变换生成， 并满足高斯分布，求出高斯分布参数
已知	样本 $X \in R^{n\times N}$
求	均值 $\mu \in R^n$、噪声方差 $\Psi \in R^{n\times n}$、线性变换 $W \in R^{n\times d}$，使得 $\max l(\mu, W, \Psi)$
解类型	迭代解
算法步骤	1. 计算均值 $\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}$ 2. 计算 B 矩阵 $B = \sum_{i=1}^{N}\mathbf{x}^{(i)}(\mathbf{x}^{(i)})^T$ 3. 随机初始化 $\Psi(0) \in R^{n\times n}$，$W(0) \in R^{n\times d}$ 4. E 步：计算下列矩阵 $E(t) = (W(t)W^T(t) + \Psi(t))^{-1}W(t)$ $C(t) = I_n - E(t)W^T(t)$ $D(t) = I_d - W^T(t)(\Psi(t) + W(t)W^T(t))^{-1}W(t)$ 5. M 步：更新参数 $W(t+1) = BE(t)(E^T(t)BE(t) + D(t))^{-1}$ $\Psi(t+1) = C(t)^T BC(t) + W(t)D(t)W^T(t)$ 6. 若似然函数收敛，停止迭代，按下式计算降维结果。否则返回 2。 $\mathbf{z} = (W^TW)^{-1}W^T(\mathbf{x} - \mu - \epsilon)$

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} \sim N\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}; \begin{bmatrix}\mu\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\Sigma_x & W\\ W^T & I_d\end{bmatrix}\right) \tag{8.24}$$

隐变量后验分布也是高斯分布，根据式2.37，有

$$\begin{aligned}\mu_{z|x} =& W^T\Sigma_x^{-1}(\mathbf{x} - \mu)\\ \Sigma_{z|x} =& I_d - W^T\Sigma_x^{-1}W\end{aligned} \tag{8.25}$$

E-M 算法的 E 步中，$p(\mathbf{x}, \mathbf{z}|W_{old}, \Psi_{old}) = p(\mathbf{z}|\mathbf{x}, W(t-1), \Psi(t-1))$，我们有

$$\begin{aligned}&\langle \log p(\mathbf{x}|\mathbf{z}, W, \Psi)\rangle_{p(\mathbf{z}|\mathbf{x}, W(t-1), \Psi(t-1))}\\ =&\left\langle -\frac{1}{2}\log|\Psi| - \frac{1}{2}(\mathbf{x} - \mu - W\mathbf{z})^T\Psi^{-1}(\mathbf{x} - \mu - W\mathbf{z}) + const\right\rangle_{p(\mathbf{z}|\mathbf{x}, W(t-1), \Psi(t-1))}\\ =&-\frac{1}{2}\log|\Psi| - \frac{1}{2}(\mathbf{x} - \mu)^T\Psi^{-1}(\mathbf{x} - \mu) - \frac{1}{2}\left\langle \mathbf{z}^TW^T\Psi^{-1}W\mathbf{z} - 2(\mathbf{x} - \mu)^T\Psi^{-1}W\mathbf{z}\right\rangle + const\\ =&-\frac{1}{2}\log|\Psi| - \frac{1}{2}(\mathbf{x} - \mu)^T\Psi^{-1}(\mathbf{x} - \mu) + (\mathbf{x} - \mu)^T\Psi^{-1}W\mu_{z|x,t-1} - \frac{1}{2}\left\langle \mathbf{z}^TW^T\Psi^{-1}W\mathbf{z}\right\rangle + const\end{aligned}$$

M 步中需要最大化 CF。对参数 W 求偏导，有

$$
\begin{aligned}
&\frac{\partial CF}{\partial W} \\
=&\frac{\partial}{\partial W}((\mathbf{x}-\mu)^T\Psi^{-1}W\mu_{z|x,t-1}-\frac{1}{2}\left\langle \mathbf{z}^TW^T\Psi^{-1}W\mathbf{z}\right\rangle) \\
=&\frac{\partial}{\partial W}\operatorname{tr}((\mathbf{x}-\mu)^T\Psi^{-1}W\mu_{z|x,t-1})-\frac{1}{2}\frac{\partial}{\partial W}\operatorname{tr}(\left\langle \mathbf{z}^TW^T\Psi^{-1}W\mathbf{z}\right\rangle) \\
=&\frac{\partial}{\partial W}\operatorname{tr}(\mu_{z|x,t-1}(\mathbf{x}-\mu)^T\Psi^{-1}W)-\frac{1}{2}\frac{\partial}{\partial W}\operatorname{tr}(\left\langle W\mathbf{z}\mathbf{z}^TW^T\Psi^{-1}\right\rangle) \\
=&\Psi^{-1}(\mathbf{x}-\mu)\mu_{z|x,t-1}^T-\Psi^{-1}W\left\langle \mathbf{z}\mathbf{z}^T\right\rangle \\
=&\Psi^{-1}(\mathbf{x}-\mu)\mu_{z|x,t-1}^T-\Psi^{-1}W(\mu_{z|x,t-1}\mu_{z|x,t-1}^T+\Sigma_{z|x,t-1})
\end{aligned}
$$

其中我们用到了式2.15的结论。考虑全数据集的梯度平均，令偏导为 0 有

$$
\sum_{i=1}^N(\mathbf{x}^{(i)}-\mu)\mu_{z|x,t-1}^T=\sum_{i=1}^NW(\mu_{z|x,t-1}^{(i)}(\mu_{z|x,t-1}^{(i)})^T+\Sigma_{z|x,t-1})
$$

因此

$$
W=\left(\sum_{i=1}^N(\mathbf{x}^{(i)}-\mu)\mu_{z|x,t-1}^T\right)\left(\sum_{i=1}^N\mu_{z|x,t-1}^{(i)}(\mu_{z|x,t-1}^{(i)})^T+\Sigma_{z|x,t-1}\right)^{-1}
$$

W 还能继续化简。根据

$$
\mu_{z|x,t-1}=W^T(t-1)\Sigma_x^{-1}(t-1)(\mathbf{x}-\mu)
$$

记

$$
B(x)=(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T
$$

有

$$
\mu_{z|x,t-1}(\mathbf{x}-\mu)^T=W^T(t-1)\Sigma_x^{-1}(t-1)B(x)
$$

考虑全数据集的梯度平均，有

$$
B=\sum_{i=1}^NB(x)=\sum_{i=1}^N\mathbf{x}^{(i)}(\mathbf{x}^{(i)})^T
$$

因此有

$$
W=\left(B\Sigma_x^{-1}W(t-1)\right)\left(W^T(t-1)\Sigma_x^{-1}B\Sigma_x^{-1}W(t-1)+\Sigma_{z|x,t-1}\right)^{-1}
$$

因此我们有参数 W 的迭代式

$$
\begin{aligned}
D(t)&=I_d-W^T(t)(\Psi(t)+W(t)W^T(t))^{-1}W(t) \\
E(t)&=(W(t)W^T(t)+\Psi(t))^{-1}W(t) \\
W(t+1)&=BE(t)(E^T(t)BE(t)+D(t))^{-1}
\end{aligned}
\tag{8.26}
$$

对参数 Ψ 求偏导，有

$$
\begin{aligned}
&\frac{\partial CF}{\partial \Psi} \\
=&\frac{\partial}{\partial \Psi}(-\frac{1}{2}\log|\Psi| - \frac{1}{2}\left\langle(\mathbf{x}-\mu-W\mathbf{z})^T\Psi^{-1}(\mathbf{x}-\mu-W\mathbf{z})\right\rangle) \\
=&-\frac{1}{2}(\frac{1}{|\Psi|}|\Psi|\Psi^{-1} - \Psi^{-1}\left\langle(\mathbf{x}-\mu-W\mathbf{z})(\mathbf{x}-\mu-W\mathbf{z})^T\Psi^{-1}\right\rangle) \\
=&\frac{1}{2}\Psi^{-1}(-I + ((\mathbf{x}-\mu)(\mathbf{x}-\mu)^T + \left\langle -2W\mathbf{z}(\mathbf{x}-\mu)^T + W\mathbf{z}\mathbf{z}^TW^T\right\rangle)\Psi^{-1}) \\
=&\frac{1}{2}\Psi^{-1}(-I + ((\mathbf{x}-\mu)(\mathbf{x}-\mu)^T - W\mu_{z|x,t-1}(\mathbf{x}-\mu)^T - (\mathbf{x}-\mu)\mu_{z|x,t-1}^TW^T \\
&+ W(\mu_{z|x,t-1}\mu_{z|x,t-1}^T + \Sigma_{z|x,t-1})W^T)\Psi^{-1})
\end{aligned}
$$

记

$$
\begin{aligned}
A =&(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T - W\mu_{z|x,t-1}(\mathbf{x}-\mu)^T - (\mathbf{x}-\mu)\mu_{z|x,t-1}^TW^T \\
&+ W(\mu_{z|x,t-1}\mu_{z|x,t-1}^T + \Sigma_{z|x,t-1})W^T
\end{aligned}
$$

令偏导为 0 有

$$
\frac{1}{2}\Psi^{-1}(-I + A\Psi^{-1}) = 0
$$

即

$$
\Psi = A
$$

矩阵 A 可以进一步化简。

$$
\begin{aligned}
A =&B(x) - WW^T(t-1)\Sigma_x^{-1}(t-1)B(x) - B(x)\Sigma_x^{-1}(t-1)W(t-1)W^T \\
&+ W(W^T(t-1)\Sigma_x^{-1}(t-1)B(x)\Sigma_x^{-1}(t-1)W(t-1) + \Sigma_{z|x,t-1})W^T \\
=&(I - WW^T(t-1)\Sigma_x^{-1}(t-1))B(x)(I - \Sigma_x^{-1}(t-1)WW^T(t-1)) + W\Sigma_{z|x,t-1}W^T
\end{aligned}
$$

综上，我们可以得到参数 Ψ 的迭代更新式

$$
\begin{aligned}
C(t) &= I_n - (\Psi(t) + W(t)W^T(t))^{-1}W(t)W^T(t) \\
D(t) &= I_d - W^T(t)(\Psi(t) + W(t)W^T(t))^{-1}W(t) \\
\Psi(t+1) &= C(t)^TBC(t) + W(t)D(t)W^T(t)
\end{aligned} \tag{8.27}
$$

9 概率图基础

概率图模型是用图来表示随机变量间依赖关系的概率模型。它和前面介绍的概率方法一脉相承，但概率图涉及到的任务、模型和数据表示等和此前的机器学习方法均有所不同，在这一章里集中说明。对概率图较为熟悉的读者可以跳过这一章。

本章和后续两章将同时涉及大量概率论和图论知识，读者如果对本书第2章列出的概念不够熟悉，需要参考其他专业教材。

符号规范：在本章和第10章、第11中，离散随机变量使用大写字母表示，如 X_i，其取值用小写字母表示，如 x_i。多个随机变量组成的随机向量用大写粗体字母表示，如 $\mathbf{X}$。随机变量的一个具体取值用小写粗体字母表示，如 $\mathbf{x}^{(i)}$ 。

对于涉及图论的部分内容，在有向图中，$Pa(X_i)$ 表示节点 X_i 的父节点集合；在无向图中，$N(X_i)$ 表示节点 X_i 的邻居节点集合。

9.1 概率图基本概念

概率图的每个**节点**代表一个随机变量。在本章和后续章节中，有时会混用“节点”和“随机变量”的概念。由于连续型随机变量的分布较为复杂，在无特殊说明的情况下，本章和后续章节中的随机变量均**默认为离散型随机变量**。使用连续性随机变量的概率图一般假定分布关系为高斯分布，因此连续随机变量概率图也称为高斯网络。

根据**边**是否有方向性，图分为有向图和无向图。概率图模型也分为**有向图模型**和**无向图模型**两类。有向图模型使用有向边表示随机变量间条件概率，而无向图模型使用无向边表示随机变量间的依赖关系。

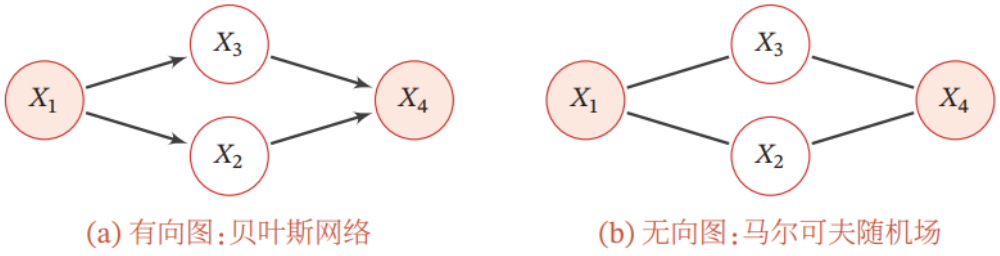

图 9.1: 有向图和无向图（图源：NNDL p.255）

有向图和无向图均可对全图随机变量的联合分布进行表示，并基于联合分布计算边缘分布、条件分布等。这一过程称为**推断**。

有向图和无向图中均有控制模型概率分布的参数。这些参数的学习过程就是概率图模型进行机器学习的过程。因此，对于概率图模型，主要有两大类任务。

推断任务：已知概率图结构及参数，求某个或某些随机变量的边缘概率或条件概率。

推断任务的复杂度会随着图模型规模的扩大而快速增加。因此有精确推断和近似推断两大类方法。精确推断方法在本章第 4 节介绍。第6章介绍的近似推断方法仍然适用于概率图模型，因此本章不多赘述。

参数学习任务：已知样本集 X、概率图结构、联合分布具体形式，求最大似然参数 $\hat{\theta}$。

除参数学习之外，对于一些应用场景，还需要学习变量之间的依赖关系，即进行图的结构学习。在大部分情况下，概率图模型的结构已经经过假设或经过领域专家设计完成，因此我们不介绍结构学习的相关内容。

9.2 有向概率图

9.2.1 基本概念

有向概率图的有向边代表节点间的条件分布。具体来说，有如下定义。

有向概率图：对于一个有向图 $G = \{V, E\}$，每个节点代点 v_i 代表一个随机变量 X_i，每条有向边代表头节点 X_i 在尾节点条件下的概率分布。对于具有多个父节点的节点 X_i，指向 X_i 的边共同组成联合条件分布。

对于一个具体的有向图，每条边的分布形式是确定的。一般对于二值变量取伯努利分布，对多值变量取多值伯努利分布。一般的二值概率图网络中，每条边的平均参数数量为 2.

联合分布：有向无环图的节点总可以进行拓扑排序。进行拓扑排序后，可以按顺序以如下公式写出全图的联合概率分布

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} p(X_i | Pa(X_i)) \tag{9.1}$$

9.2.2 条件独立性

对于有向概率图模型，有如下三种局部节点连接关系：间接因果关系、共因关系、共果关系。它们分别对应图9.2中的 (a) 与 (b)，(c)，(d)。对于这些局部连接关系，有如下节点（条件）独立性质。

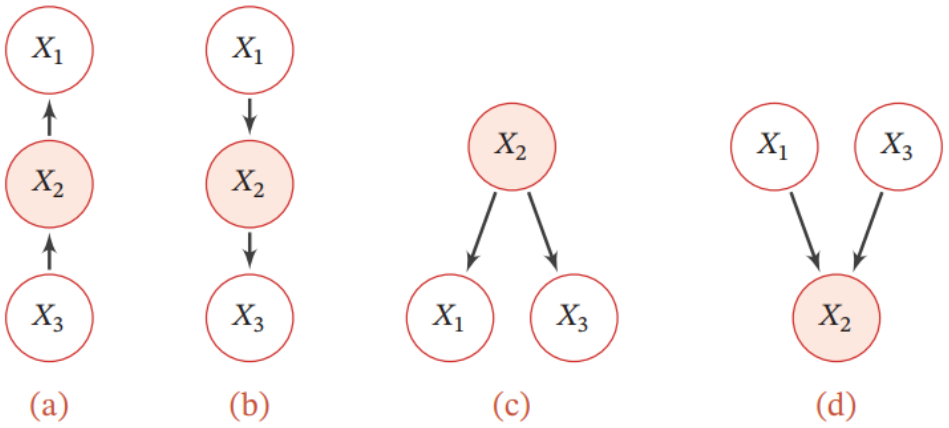

图 9.2: 有向图的局部连接关系（图源：NNDL p.256）

- 间接因果关系：图9.2(a)(b) 中，$X_1 \perp X_3 | X_2$。
- 共因关系：图9.2(c) 中，$X_1 \perp X_3 | X_2$。
- 共果关系：图9.2(d) 中，$X_1 \perp X_3 | \varnothing$。

证明. **间接因果关系**（以图9.2(a) 为例）

根据式9.1，写出联合分布形式

$$p(X_1, X_2, X_3) = p(X_3)p(X_2|X_3)p(X_1|X_2)$$

因此有

$$
\begin{aligned}
p(X_1|X_2)p(X_3|X_2) =& \frac{P(X_1,X_2)P(X_3,X_2)}{P(X_2)^2} \\
=& \frac{(\sum_{X_3} p(X_1,X_2,X_3))(\sum_{X_1} p(X_1,X_2,X_3))}{P(X_2)^2} \\
=& \frac{(p(X_1|X_2)\sum_{X_3} p(X_3)p(X_2|X_3))(p(X_3)p(X_2|X_3)\sum_{X_1} p(X_1|X_2))}{P(X_2)^2} \\
=& \frac{p(X_1|X_2)p(X_2)p(X_3)p(X_2|X_3)}{P(X_2)^2} \\
=& \frac{p(X_1|X_2)p(X_3)p(X_2|X_3)}{P(X_2)} \\
=& \frac{p(X_1,X_2,X_3)}{P(X_2)} \\
=& p(X_1,X_3|X_2)
\end{aligned}
$$

共因关系

联合分布形式

$$
p(X_1,X_2,X_3) = p(X_2)p(X_1|X_2)p(X_3|X_2)
$$

因此有

$$
\begin{aligned}
p(X_1|X_2)p(X_3|X_2) =& \frac{P(X_1,X_2)P(X_3,X_2)}{P(X_2)^2} \\
=& \frac{(\sum_{X_3} p(X_1,X_2,X_3))(\sum_{X_1} p(X_1,X_2,X_3))}{P(X_2)^2} \\
=& \frac{(p(X_2)p(X_1|X_2)\sum_{X_3} p(X_3|X_2))(p(X_2)p(X_3|X_2)\sum_{X_1} p(X_1|X_2))}{P(X_2)^2} \\
=& \frac{p(X_2)p(X_1|X_2)p(X_2)p(X_3|X_2)}{P(X_2)^2} \\
=& \frac{p(X_1|X_2)p(X_2)p(X_3|X_2)}{P(X_2)} \\
=& \frac{p(X_1,X_2,X_3)}{P(X_2)} \\
=& p(X_1,X_3|X_2)
\end{aligned}
$$

共果关系

联合分布形式

$$
p(X_1,X_2,X_3) = p(X_1)p(X_3)p(X_2|X_1,X_3)
$$

因此有

$$
\begin{aligned}
p(X_1, X_3) &= \sum_{X_2} p(X_1, X_2, X_3) \\
&= \sum_{X_2} p(X_1)p(X_3)p(X_2|X_1, X_3) \\
&= p(X_1)p(X_3) \sum_{X_2} p(X_2|X_1, X_3) \\
&= p(X_1)p(X_3)
\end{aligned}
$$

□

9.3 无向概率图

9.3.1 基本概念

局部马尔可夫性：若无向图 $G = \{V, E\}$ 中的节点 v_i 代表随机变量 X_i，且随机变量间的关系满足

$$
p(X_i|X_{-i}) = p(X_i|N(X_i)) \tag{9.2}
$$

则称图 G 满足局部马尔可夫性。此时称图 G 是一个**马尔科夫随机场**（MRF）。按照惯例，也称图 G 是一个无向概率图模型。

MRF 中的一个基石性定理是 **Hammersley-Clifford 定理**，它的内容是：

Theorem 9.1. *一组随机变量 $\mathbf{X}$ 及无向图 $G = \{V, E\}$ 满足局部马尔可夫性质，当且仅当 $p(\mathbf{X})$ 可以表示为一系列定义在最大团上的非负函数的乘积形式，即式9.3。*

$$
p(\mathbf{X}) = \frac{1}{Z} \prod_{i=1}^{K} \phi_{C_i}(\mathbf{X}_{C_i}) \tag{9.3}
$$

式9.3中，C_i 表示图 G 上的第 i 个最大团。ϕ_{C_i} 是定义在这个团上的非负函数，称为**势函数**。Z 是势函数的归一化因子，也称为**配分函数**，定义为式9.4。其中 $\sum_{\mathbf{x}}$ 表示对 x 定义域上所有取值求和。若 $\mathbf{x}$ 是 n 维 $\{0\ 1\}$ 二值变量，则 $\mathbf{x}$ 的取值有 2^n 种可能。

$$
Z = \sum_{\mathbf{x}} \prod_{i=1}^{K} \phi_{C_i}(\mathbf{x}_{C_i}) \tag{9.4}
$$

注：并不是所有形式的非负函数都能成为势函数。只有配分函数存在，非负函数乘积求和能被归一化为概率分布，才是有意义的势函数。

Hammersley-Clifford 定理不仅指出了无向图模型的联合概率分布形式，还揭示了无向图中边的含义：边的势函数之积就是未归一化的联合概率分布。不过，势函数并不是概率分布。有向图的边明确代表着条件概率，而无向图的边的势函数和两端节点联合分布的关系并不明显。

在实际使用中，具体的图模型会定义具体的势函数，并且会有特定的图结构，以方便各种概率的计算和参数学习。

9.3.2 条件独立性

无向图具有基于分离的条件独立性。对于无向图中的三个节点集 A, B, C，若任意从 A 中节点到 B 中节点的路径都要经过 C 中的节点，称节点集 C 将 A 和 B **分离**。

Theorem 9.2. *对于无向图 G，若节点集 C 将 A 和 B 分离，则有 A 和 B 关于 C 条件独立，即式9.5*

$$p(A, B|C) = p(A|C)p(B|C) \tag{9.5}$$

无向图的条件独立性、局部马尔可夫性和势函数的定义是互相等价的。出于篇幅考虑不在这里证明，读者可以自行查阅相关文献。

9.4 精确推断方法

理论上，一个概率图模型只要给定模型的定义和参数，就可以求出图上任何变量之间的联合分布、条件分布或变量的边缘分布。然而，无论是有向图还是无向图，根据定义直接求解的计算量都是很大的。因此，有一些简化概率图分布计算的算法。由于这些算法都是精确计算概率分布，因此称为精确推断方法。相对的还有后续介绍的近似推断方法。

（待完善）本节主要介绍三种精确推断方法：变量消除算法、信念传播算法和 Max Product 算法。

9.5 近似推断方法

上述精确推断方法可以有效解决概率图推断的问题，但这一切都有一个前提——节点中没有隐变量。然而，很多机器学习方法恰恰是基于含有隐变量的概率图模型的。通过假设隐变量节点和观测节点（样本对应的随机变量）之间的连接关系，就可以建模含有隐变量的概率模型 $p(\mathbf{X}, \mathbf{Z}|\theta)$，让模型拥有更强的表示能力，恰如我们在第8章介绍的 GMM 等模型一样。

针对含有隐变量的参数学习问题，我们在第6章已经介绍了 E-M 算法和一些近似方法。但对于每种概率图模型，针对其自身的结构可以开发特定的近似方法，如 Sigmoid 信念网络的醒眠算法、RBM 的对比散度算法等等。具体请参见下一章的介绍。

10 二值概率图模型

本章将介绍一系列含有**隐节点**的二值概率图模型。隐节点是指没有观测数据的概率图节点，相对而言对应于数据集的随机变量称为**观测节点**。

尽管这些图模型也可用于多值或连续随机变量，但为简单起见，以下推导中我们仅考虑二值模型，即每个隐节点和观测节点都是二值的。对于观测节点，现有文献往往记作 $\mathbf{v}$。本书中为了和前面的符号规范相统一，仍记作 $\mathbf{x}$。

10.1 Sigmoid 信念网络

Sigmoid 信念网络 (SBN) 是一种含有隐变量（隐节点）的有向概率图模型，是一种生成式模型，常用于无监督机器学习任务。

具体来说，SBN 是一个 L 层神经网络，第 l 层的隐节点有向连接到第 $l-1$ 层，第 0 层为观测节点。每层隐节点之间互不连接，观测节点间也互不连接，但每两层节点是全连接关系，如图10.1所示。

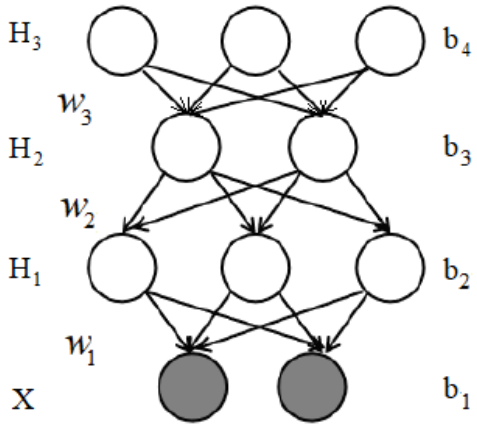

图 10.1: Sigmoid 信念网络结构示意

假设第 l 层的节点为 $\mathbf{H}_l$，第 l 层的第 i 个节点为 $H_{l,i}$，定义节点取 1 的后验概率为

$$p(H_{l,i}=1|\mathbf{H},\theta)=\text{sig}(\mathbf{w}_{l+1,i}^T\mathbf{H}_{l+1}+\mathbf{b}_{l+1}) \tag{10.1}$$

其中 $\mathbf{w}_{l+1,i}\in R^{n_{l+1}},\mathbf{b}_{l+1}\in R^{n_{l+1}}$ 是参数。n_{l+1} 是第 $l+1$ 层的节点数。记 $\theta=\{\mathbf{w},\mathbf{b}\}$。注意最高层节点的分布不受其他节点控制。由于

$$p(H_{l,i}=0|\mathbf{H},\theta)=1-p(H_{l,i}=0|\mathbf{H},\theta)=\text{sig}(-\mathbf{w}_{l+1}^T\mathbf{H}_{l+1}-\mathbf{b}_{l+1})$$

也可以将其写为

$$p(H_{l,i}|\mathbf{H},\theta)=\text{sig}((2H_{l,i}-1)(\mathbf{w}_{l+1}^T\mathbf{H}_{l+1}+\mathbf{b}_{l+1}))$$

于是网络全部节点的联合分布为

$$p(\mathbf{H},\mathbf{X}|\theta)=\prod_{l=0}^{L}\prod_{i=1}^{n_l}p(H_{l,i}|\mathbf{H},\theta) \tag{10.2}$$

其中取 $\mathbf{H}_0 = \mathbf{X}$。

10.1.1 SBN 梯度上升

取似然函数的对数值为优化目标，有

$$\max_{\theta} \log p(\mathbf{X}|\theta) = \log \sum_{H} p(\mathbf{H}, \mathbf{X}|\theta)$$

由于形式复杂，考虑对其进行梯度上升迭代。求导数有

$$\begin{aligned}
&\frac{\partial}{\partial \theta} \log p(\mathbf{X}|\theta) \\
=&\frac{\frac{\partial}{\partial \theta} p(\mathbf{X}|\theta)}{p(\mathbf{X}|\theta)} \\
=&\frac{\frac{\partial}{\partial \theta} \sum_{\mathbf{H}} p(\mathbf{X}, \mathbf{H}|\theta)}{p(\mathbf{X}|\theta)} \\
=&\sum_{\mathbf{H}} \frac{\frac{\partial}{\partial \theta} p(\mathbf{X}, \mathbf{H}|\theta)}{p(\mathbf{X}|\theta)} \\
=&\sum_{\mathbf{H}} p(\mathbf{H}|\mathbf{X}, \theta) \frac{\frac{\partial}{\partial \theta} p(\mathbf{X}, \mathbf{H}|\theta)}{p(\mathbf{X}, \mathbf{H}|\theta)}
\end{aligned}$$

具体到参数，有

$$\begin{aligned}
&\frac{\partial}{\partial w_{l+1,i,j}} \log p(\mathbf{X}|\theta) \\
=&\sum_{\mathbf{H}} p(\mathbf{H}|\mathbf{X}, \theta) \frac{\frac{\partial}{\partial w_{l+1,i,j}} p(\mathbf{X}, \mathbf{H}|\theta)}{p(\mathbf{X}, \mathbf{H}|\theta)} \\
=&\sum_{\mathbf{H}} p(\mathbf{H}|\mathbf{X}, \theta) \frac{\frac{\partial}{\partial w_{l+1,i,j}} \prod_{l'=0}^{L} \prod_{i'=1}^{n_{l'}} p(H_{l',i'}|\mathbf{H}, \theta)}{\prod_{l'=0}^{L} \prod_{i'=1}^{n_{l'}} p(H_{l',i'}|\mathbf{H}, \theta)} \\
=&\sum_{\mathbf{H}} p(\mathbf{H}|\mathbf{X}, \theta) \frac{p(H_{-l,-i}|\mathbf{H}, \theta) \frac{\partial}{\partial w_{l+1,i,j}} p(H_{l,i}|\mathbf{H}, \theta)}{p(H_{-l,-i}|\mathbf{H}, \theta) p(H_{l,i}|\mathbf{H}, \theta)} \\
=&\sum_{\mathbf{H}} p(\mathbf{H}|\mathbf{X}, \theta) \frac{\frac{\partial}{\partial w_{l+1,i,j}} p(H_{l,i}|\mathbf{H}, \theta)}{p(H_{l,i}|\mathbf{H}, \theta)} \\
=&\sum_{\mathbf{H}} p(\mathbf{H}|\mathbf{X}, \theta) \frac{\frac{\partial}{\partial w_{l+1,i,j}} p(H_{l,i}|\mathbf{H}, \theta)}{p(H_{l,i}|\mathbf{H}, \theta)}
\end{aligned}$$

其中 $p(H_{-l,-i}|\mathbf{H}, \theta)$ 表示除第 l 层第 i 个隐变量外的所有隐变量的条件联合分布。记

$$s_{l,i} = (2H_{l,i} - 1)(\mathbf{w}_{l+1}^T \mathbf{H}_{l+1} + \mathbf{b}_{l+1}) \tag{10.3}$$

则有

$$
\begin{aligned}
&\frac{\frac{\partial}{\partial w_{l+1,i,j}} p(H_{l,i}|\mathbf{H},\theta)}{p(H_{l,i}|\mathbf{H},\theta)} \\
=&\frac{\frac{\partial}{\partial w_{l+1,i,j}} \operatorname{sig}(s_{l,i})}{\operatorname{sig}(s_{l,i})} \\
=&\frac{\operatorname{sig}(s_{l,i})\operatorname{sig}(-s_{l,i})(2H_{l,i}-1)\mathbf{H}_{l+1,j}}{\operatorname{sig}(s_{l,i})} \\
=&\operatorname{sig}(-s_{l,i})(2H_{l,i}-1)\mathbf{H}_{l+1,j}
\end{aligned}
$$

因此，梯度为

$$
\frac{\partial}{\partial w_{l+1,i,j}} \log p(\mathbf{X}|\theta) = \langle \operatorname{sig}(-s_{l,i})(2H_{l,i}-1)\mathbf{H}_{l+1,j} \rangle_{p(\mathbf{H}|\mathbf{X},\theta)}
$$

同理有

$$
\frac{\partial L}{\partial b_{l+1,i}} \log p(\mathbf{X}|\theta) = \langle \operatorname{sig}(-s_{l,i})(2H_{l,i}-1) \rangle_{p(\mathbf{H}|\mathbf{X},\theta)}
$$

可以看到，参数梯度依赖于隐变量的条件分布。由于第 1 层隐变量和观测变量是共果连接关系，给定观测变量时隐变量并不独立，条件分布非常难求。只有节点数量较少时可以容易求解。这使得 SBN 的最大似然优化非常困难。

10.1.2 醒眠算法

上一小节中我们看到，SBN 需要优化对数似然 $\log L$，而隐变量后验 $p(\mathbf{z}|\mathbf{x},\theta)$ 很不好求。根据我们之前的经验，一个自然的想法是使用另一个较为简单的分布 $q(\mathbf{z}|\mathbf{x},\phi)$ 来近似实际后验分布。我们在变分推断里使用平均场假设将后验分解。在这里我们使用另一个思路：使用**反向网络**，即另一个连接相同，但每条边方向相反的概率图网络来近似 SBN 的隐变量后验。即：

$$
p(H_{l+1,i}=1|\mathbf{X},\theta) \approx q(H_{l+1,i}=1|\mathbf{X},\phi) = \operatorname{sig}(\mathbf{r}_{l,i}^T\mathbf{H}_l + \mathbf{c}_l)
$$

其中 $\mathbf{r}_{l,i} \in R^{n_l}, \mathbf{c}_l \in R^{n_{l+1}}$ 是参数。记 $\phi = \{\mathbf{r}, \mathbf{c}\}$。因此我们有梯度近似

$$
\begin{aligned}
\frac{\partial}{\partial w_{l+1,i,j}} \log p(\mathbf{X}|\theta) &\approx \langle \operatorname{sig}(-s_{l,i})(2H_{l,i}-1)\mathbf{H}_{l+1,j} \rangle_{q(\mathbf{H}|\mathbf{X},\phi)} \\
\frac{\partial L}{\partial b_{l+1,i}} \log p(\mathbf{X}|\theta) &\approx \langle \operatorname{sig}(-s_{l,i})(2H_{l,i}-1) \rangle_{q(\mathbf{H}|\mathbf{X},\phi)}
\end{aligned}
$$

也就是说，我们在获得一组 ϕ 的情况下，根据样本直接从 $q(\mathbf{H}|\mathbf{X},\phi)$ 中采样，就可以得到近似的梯度值。这组梯度值可以用于一次梯度下降的参数更新。由于隐节点全部为 $\{0,1\}$ 变量，服从参数为 sigmoid 的伯努利分布，采样是比较简单的，只需要由下到上逐层计算即可。

那么，如何获得一组 ϕ 呢？注意到 $q(\mathbf{H}|\mathbf{X},\phi)$ 组成的概率图也是一个 SBN（我们可以将 $\mathbf{H}_L$ 层节点看作新网络的“观测节点” $\mathbf{V}$），而 $p(\mathbf{z}|\mathbf{x},\theta)$ 正是这一 SBN 的后验近似。因此完全类似地，我们用 $p(\mathbf{z}|\mathbf{x},\theta)$ 中抽样的“样本”更新 $q(\mathbf{H}|\mathbf{X},\phi)$ 的参数。

算法	**Sigmoid 信念网络-醒眠算法**
算法简述	对于 Sigmoid 信念网络的学习问题，利用反向网络进行近似梯度上升迭代求解
已知	样本 $\mathbf{x}^{(i)} \in R^{n\times N}, i=1,2,...,N$，SBN 网络及反向网络结构，学习率 α
求	参数 $\mathbf{w},\mathbf{b}$ 的最大似然解
解类型	迭代解
算法步骤	1. 随机初始化参数 $\mathbf{w},\mathbf{b}$ 和 $\mathbf{r},\mathbf{c}$ 2. (Wake-step) 对每一样本 $\mathbf{x}^{(k)}$，根据反向网络条件概率采样隐变量 $\mathbf{h}^{(k)}$ $h_{l+1,i} \sim q(h_{l+1,i}\|\mathbf{x},\phi(t)) = (2h_{l+1,i}-1)\,\mathrm{sig}(\mathbf{r}_{l,i}^T(t)\mathbf{h}_l + \mathbf{c}_l(t))$ 3. (Wake-step) 按下式计算参数梯度并更新 $s_{l,i}^{(k)} = (2h_{l,i}^{(k)}-1)(\mathbf{w}_{l+1}^T(t)\mathbf{h}_{l+1}^{(k)} + \mathbf{b}_{l+1}^{(k)}(t))$ $\frac{\partial L}{\partial w_{l+1,i,j}} \approx \frac{1}{N}\sum_{k=1}^N \mathrm{sig}(-s_{l,i}^{(k)})(2h_{l,i}^{(k)}-1)\mathbf{h}_{l+1,j}^{(k)}$ $\frac{\partial L}{\partial b_{l+1,i}} \approx \frac{1}{N}\sum_{k=1}^N \mathrm{sig}(-s_{l,i}^{(k)})(2h_{l,i}^{(k)}-1)$ $\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\frac{\partial L}{\partial \mathbf{w}}$ $\mathbf{b}(t+1) = \mathbf{b}(t) + \alpha\frac{\partial L}{\partial \mathbf{b}}$ 4. (Sleep-step) 取反向观测样本 $\mathbf{v}^{(k)} = \mathbf{h}_l^{(k)}$， 根据反向网络条件概率采样隐变量 $\mathbf{h}'^{(k)}$ $h'_{l,i} \sim p(h_{l,i}\|\mathbf{h},\theta) = \mathrm{sig}((2h_{l,i}-1)(\mathbf{w}_{l+1}^T(t+1)\mathbf{H}_{l+1} + \mathbf{b}_{l+1}(t+1)))$ 5. (Sleep-step) 按下式计算反向网络参数梯度并更新 $s'^{(k)}_{l,i} = (2h'^{(k)}_{l,i}-1)(\mathbf{r}_l^T(t)\mathbf{h}'^{(k)}_l + \mathbf{c}_l^{(k)}(t))$ $\frac{\partial L}{\partial r_{l+1,i,j}} \approx \frac{1}{N}\sum_{k=1}^N \mathrm{sig}(-s'^{(k)}_{l,i})(2h'^{(k)}_{l+1,i}-1)\mathbf{h}'^{(k)}_{l,j}$ $\frac{\partial L}{\partial c_{l+1,i}} \approx \frac{1}{N}\sum_{k=1}^N \mathrm{sig}(-s'^{(k)}_{l,i})(2h'^{(k)}_{l+1,i}-1)$ $\mathbf{r}(t+1) = \mathbf{r}(t) + \alpha\frac{\partial L}{\partial \mathbf{r}}$ $\mathbf{c}(t+1) = \mathbf{c}(t) + \alpha\frac{\partial L}{\partial \mathbf{c}}$ 6. 若似然函数收敛，停止迭代，计算降维结果。否则返回 2。

$$\frac{\partial L'}{\partial r_{l,i,j}}\log q(\mathbf{X}'|\phi) \approx \left\langle \mathrm{sig}(-s'_{l,i})(2H_{l+1,i}-1)\mathbf{H}_{l,j}\right\rangle_{p(\mathbf{H}|\mathbf{X}',\phi)}$$
$$\frac{\partial L'}{\partial c_{l,i}}\log p(\mathbf{X}'|\theta) \approx \left\langle \mathrm{sig}(-s'_{l,i})(2H_{l+1,i}-1)\right\rangle_{p(\mathbf{H}|\mathbf{X}',\phi)}$$

这样，我们可以交替更新 $q(\mathbf{z}|\mathbf{x},\phi)$ 和 $p(\mathbf{x}|\mathbf{z},\theta)$ 的参数。这很类似于 E-M 算法和 CAVI 的思想。

需要注意的一点是：醒眠算法只是在实践中被证明可以用于优化参数，并没有理论保证其收敛性。

10.2 玻尔兹曼机 BM

玻尔兹曼机是一类含隐节点的全连接无向概率图模型，如图10.2所示。假设隐变量个数为 m，观测变量个数为 n，定义其能量函数和势函数为

$$E(\mathbf{X},\mathbf{H}) = -(\mathbf{H}^T W\mathbf{X} + \frac{1}{2}\mathbf{H}^T J\mathbf{H} + \frac{1}{2}\mathbf{X}^T K\mathbf{X}) \tag{10.4}$$

$$\phi(\mathbf{X}, \mathbf{H}) = \exp\{-E(\mathbf{X}, \mathbf{H})\} \tag{10.5}$$

其中 $W \in R^{m\times n}, J \in R^{m\times m}, L \in R^{n\times n}$ 是模型参数。则有联合概率分布

$$p(\mathbf{X}, \mathbf{H}) = \frac{1}{Z}\exp\{-E(\mathbf{X}, \mathbf{H})\} = \frac{1}{Z}\exp\{\mathbf{H}^T W\mathbf{X} + \frac{1}{2}\mathbf{H}^T J\mathbf{H} + \frac{1}{2}\mathbf{X}^T K\mathbf{X}\} \tag{10.6}$$

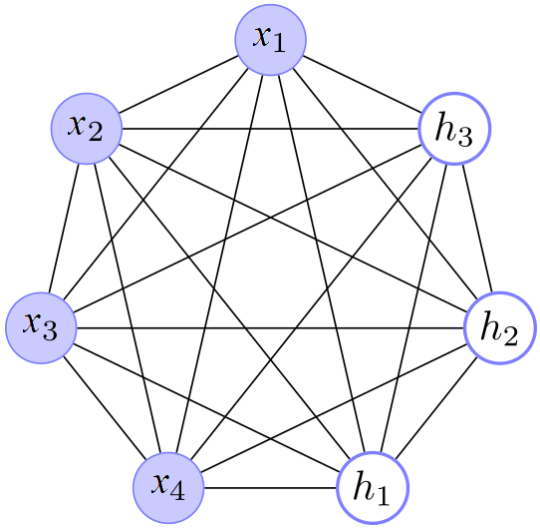

图 10.2: 玻尔兹曼机结构示意

10.2.1 BM 梯度上升

玻尔兹曼机是一种生成模型，其参数应当使得观测变量的边缘概率，也就是参数似然最大，即

$$\max p(\mathbf{x}; W, J, K)$$

这里我们暂且先只考虑一个样本 $\mathbf{x}$。最大化似然相当于最大化对数似然，即

$$\max L(W, J, K) = \log p(\mathbf{x}; W, J, K),$$

为简便起见，记 $\theta = \{W, J, K\}$，以下所有边缘概率、条件概率、联合概率均不显示的写出以参数为条件。由于

$$p(\mathbf{x}) = \frac{\sum_{\mathbf{h}} \phi(\mathbf{x}, \mathbf{h})}{\sum_{\mathbf{H}}\sum_{\mathbf{X}} \phi(\mathbf{X}, \mathbf{H})}$$

有对数似然表达式

$$L(W, J, L) = \log p(\mathbf{x}) = \log \sum_{\mathbf{h}} \phi(\mathbf{x}, \mathbf{h}) - \log \sum_{\mathbf{H}}\sum_{\mathbf{X}} \phi(\mathbf{X}, \mathbf{H})$$

为了最大化对数似然，考虑对其求偏导

算法	**玻尔兹曼机-梯度上升**
算法简述	对于玻尔兹曼机的学习问题，进行最大似然梯度上升的参数迭代求解
已知	样本 $\mathbf{x}^{(i)} \in R^{n\times N}, i=1,2,...,N$，BM 网络结构，学习率 α， Gibbs 采样隐变量条件分布最少收敛次数 S_1，联合分布最少收敛次数 S_2
求	参数 W, J, K 的最大似然解
解类型	迭代解
算法步骤	1. 随机初始化参数 $W, \mathbf{a}, \mathbf{b}$ 2. 对每一样本，随机初始化一组隐变量，按单个隐变量条件分布对 每个隐变量轮流采样至少 S_1 轮，得到 $\mathbf{h}_C^{(k)}$ $p(h_i=1\|\mathbf{h}_{-i},\mathbf{x})=\text{sig}(W_{i,:}\mathbf{x}+\frac{1}{2}J_{i,i}+J_i^T\mathbf{h}_{-i})$ $p(x_i=1\|\mathbf{x}_{-i},\mathbf{h})=\text{sig}(\mathbf{h}^TW_{:,i}+\frac{1}{2}K_{i,i}+\mathbf{h}_{-i}^TK_i)$ 3. 对每一样本，随机初始化一组隐变量，按单个隐变量条件分布对 每个隐变量和观测变量轮流采样至少 S_2 轮，得到 $\mathbf{x}_J^{(k)}, \mathbf{h}_J^{(k)}$ 4. 按下式计算各参数梯度 $\frac{\partial L}{\partial W}=\frac{1}{N}\sum_{k=1}^N\left(\mathbf{h}_J^{(k)}(\mathbf{x}_J^{(k)})^T-\mathbf{h}_C^{(k)}(\mathbf{x}^{(k)})^T\right)$ $\frac{\partial L}{\partial J}=\frac{1}{N}\sum_{k=1}^N\left(\mathbf{h}_J^{(k)}(\mathbf{h}_J^{(k)})^T-\mathbf{h}_C^{(k)}(\mathbf{h}_C^{(k)})^T\right)$ $\frac{\partial L}{\partial K}=\frac{1}{N}\sum_{k=1}^N\left(\mathbf{x}_J^{(k)}(\mathbf{x}_J^{(k)})^T-\mathbf{x}^{(k)}(\mathbf{x}^{(k)})^T\right)$ 3. 进行梯度上升参数更新 $\theta(t+1)=\theta(t)+\alpha\frac{\partial L}{\partial\theta}$ 4. 若似然函数收敛，停止迭代，否则返回 2。

$$
\begin{aligned}
&\frac{\partial L}{\partial\theta}=\frac{\partial}{\partial\theta}\log\sum_{\mathbf{h}}\phi(\mathbf{x},\mathbf{h})-\frac{\partial}{\partial\theta}\log\sum_{\mathbf{H}}\sum_{\mathbf{X}}\phi(\mathbf{X},\mathbf{H})\\
=&\frac{1}{\sum_{\mathbf{h}}\phi(\mathbf{x},\mathbf{h})}\sum_{\mathbf{h}}\frac{\partial}{\partial\theta}\phi(\mathbf{x},\mathbf{h})-\frac{1}{\sum_{\mathbf{H}}\sum_{\mathbf{X}}\phi(\mathbf{X},\mathbf{H})}\sum_{\mathbf{H}}\sum_{\mathbf{X}}\frac{\partial}{\partial\theta}\phi(\mathbf{X},\mathbf{H})
\end{aligned}
$$

由式10.5，有

$$
\frac{\partial L}{\partial\theta}\phi(\mathbf{x},\mathbf{h})=-\phi(\mathbf{x},\mathbf{h})\frac{\partial L}{\partial\theta}E(\mathbf{x},\mathbf{h})
$$

因此

$$
\begin{aligned}
&\frac{\partial L}{\partial\theta}\\
=&\frac{-\sum_{\mathbf{h}}\phi(\mathbf{x},\mathbf{h})\frac{\partial L}{\partial\theta}E(\mathbf{x},\mathbf{h})}{\sum_{\mathbf{h}}\phi(\mathbf{x},\mathbf{h})}-\frac{-\sum_{\mathbf{H}}\sum_{\mathbf{X}}\phi(\mathbf{X},\mathbf{H})\frac{\partial L}{\partial\theta}E(\mathbf{X},\mathbf{H})}{\sum_{\mathbf{H}}\sum_{\mathbf{X}}\phi(\mathbf{X},\mathbf{H})}\\
=&-\sum_{\mathbf{h}}p(\mathbf{h}|\mathbf{x})\frac{\partial L}{\partial\theta}E(\mathbf{x},\mathbf{h})+\sum_{\mathbf{H}}\sum_{\mathbf{X}}p(\mathbf{X},\mathbf{H})\frac{\partial L}{\partial\theta}E(\mathbf{X},\mathbf{H})\\
=&-\left\langle\frac{\partial L}{\partial\theta}E(\mathbf{x},\mathbf{h})\right\rangle_{p(\mathbf{h}|\mathbf{x})}+\left\langle\frac{\partial L}{\partial\theta}E(\mathbf{X},\mathbf{H})\right\rangle_{p(\mathbf{X},\mathbf{H})}
\end{aligned}
$$

于是我们有结论

$$
\frac{\partial L}{\partial\theta}=-\left\langle\frac{\partial L}{\partial\theta}E(\mathbf{x},\mathbf{h}|\theta)\right\rangle_{p(\mathbf{h}|\mathbf{x},\theta)}+\left\langle\frac{\partial L}{\partial\theta}E(\mathbf{X},\mathbf{H}|\theta)\right\rangle_{p(\mathbf{H},\mathbf{X}|\theta)}\tag{10.7}
$$

注意，该结论并不涉及具体的能量函数形式，适用于任何含有隐变量的无向图模型。具体到玻尔兹曼机的模型参数，有

$$\begin{aligned}\frac{\partial L}{\partial W} &= -\left\langle \mathbf{h}\mathbf{x}^T \right\rangle_{p(\mathbf{h}|\mathbf{x})} + \left\langle \mathbf{H}\mathbf{X}^T \right\rangle_{p(\mathbf{H},\mathbf{X}|\theta)} \\ \frac{\partial L}{\partial J} &= -\left\langle \mathbf{h}\mathbf{h}^T \right\rangle_{p(\mathbf{h}|\mathbf{x})} + \left\langle \mathbf{H}\mathbf{H}^T \right\rangle_{p(\mathbf{H},\mathbf{X}|\theta)} \\ \frac{\partial L}{\partial K} &= -\left\langle \mathbf{x}\mathbf{x}^T \right\rangle_{p(\mathbf{h}|\mathbf{x})} + \left\langle \mathbf{X}\mathbf{X}^T \right\rangle_{p(\mathbf{H},\mathbf{X}|\theta)}\end{aligned}$$

这里，所有的梯度都分为两部分，我们一般把前一部分称为“负相”，后一部分称为“正相”。正相要对隐变量和观测变量联合分布求期望，而负相要对给定观测变量条件下隐变量联合分布求期望。遗憾的是，它们大都无法解析求解，只能考虑使用近似方法。

回顾第6章介绍的 Gibbs 采样，对于联合分布 $p(\mathbf{H},\mathbf{X}|\theta)$，如果我们可以计算出 $p(H_i|\mathbf{H}_{-i},\mathbf{X})$，我们就可以对各隐变量和观测变量依次采样，在达到平稳分布后认为样本服从联合分布。对于条件分布，我们也可以根据 $p(h_i|\mathbf{h}_{-i},\mathbf{x})$ 进行轮流采样，在达到平稳分布后认为样本服从条件分布。

具体来说，我们有

$$p(h_i = 1|\mathbf{h}_{-i},\mathbf{x}) = \frac{p(h_i = 1,\mathbf{h}_{-i},\mathbf{x})}{p(h_i = 1,\mathbf{h}_{-i},\mathbf{x}) + p(h_i = 0,\mathbf{h}_{-i},\mathbf{x})}$$

记 W 去掉第 i 行的矩阵为 $W_{-i,:}$，去掉第 j 列的矩阵为 $W_{:,-j}$。记 J,K 分别去掉第 i 行和第 i 列后的矩阵为 J_{-i},K_{-i}，记 J,K 的第 i 列除去第 i 个元素后为 J_i,K_i，记

$$\phi(\mathbf{h}_{-i},\mathbf{x}) = \exp\{\mathbf{h}_{-i}^T W_{-i,:}\mathbf{x} + \frac{1}{2}\mathbf{h}_{-i}^T J_{-i}\mathbf{h}_{-i} + \frac{1}{2}\mathbf{x}^T K\mathbf{x}\}$$

则有

$$\begin{aligned}&p(h_i = 1|\mathbf{h}_{-i},\mathbf{x}) \\ =&\frac{p(h_i = 1,\mathbf{h}_{-i},\mathbf{x})}{p(h_i = 1,\mathbf{h}_{-i},\mathbf{x}) + p(h_i = 0,\mathbf{h}_{-i},\mathbf{x})} \\ =&\frac{\frac{1}{Z}\phi(\mathbf{h}_{-i},\mathbf{x})\exp\{W_{i,:}\mathbf{x} + \frac{1}{2}J_{i,i} + J_i^T\mathbf{h}_{-i}\}}{\frac{1}{Z}\phi(\mathbf{h}_{-i},\mathbf{x})\exp\{W_{i,:}\mathbf{x} + \frac{1}{2}J_{i,i} + J_i^T\mathbf{h}_{-i}\} + \frac{1}{Z}\phi(\mathbf{h}_{-i},\mathbf{x})} \\ =&\frac{\exp\{W_{i,:}\mathbf{x} + \frac{1}{2}J_{i,i} + J_i^T\mathbf{h}_{-i}\}}{\exp\{W\mathbf{x} + \frac{1}{2}J_{i,i} + J_i^T\mathbf{h}_{-i}\} + 1} \\ =&\mathrm{sig}(W_{i,:}\mathbf{x} + \frac{1}{2}J_{i,i} + J_i^T\mathbf{h}_{-i})\end{aligned}$$

同理我们有

$$p(x_i = 1|\mathbf{x}_{-i},\mathbf{h}) = \mathrm{sig}(\mathbf{h}^T W_{:,i} + \frac{1}{2}K_{i,i} + \mathbf{h}_{-i}^T K_i)$$

因此，我们可以利用这两式进行 Gibbs 采样，完成上面的梯度计算。

10.2.2 BM 变分推断

上面的梯度上升算法中，每轮参数更新都需要进行 2 次 Gibbs 采样，每次采样都需要等待很长时间。这使得参数学习的效率很低。这很大程度上是因为隐变量后验 $p(\mathbf{h}|\mathbf{x})$ 无法求出。如果能对隐变量后验进行某种近似，从近似分布中进行抽样，就可以缓解这一问题。

算法	**玻尔兹曼机-变分推断**
算法简述	对于玻尔兹曼机的隐变量后验分布估计问题，使用变分推断迭代求解
已知	样本 $\mathbf{x}^{(i)} \in R^{n\times N}, i = 1, 2, ..., N$，BM 网络结构
求	变分近似 $q(h_i|\mathbf{x}, \phi_i) = \phi_i^{h_i}(1-\phi_i)^{1-h_i}$ 的最优参数 $\phi^\star$
解类型	近似解
算法步骤	1. 随机初始化参数 $\phi \in R^m$ 2. 按下式轮流更新 ϕ_k：$\phi_k = \text{sig}(\sum_{i=0}^{n} w_{k,i}x_i + \sum_{j=0}^{m} j_{k,j}\phi_j)$ 其中 $\mathbf{x} = \frac{1}{N}\sum_{l=1}^{N}\mathbf{x}^{(l)}$ 3. 若参数收敛，停止迭代，否则返回 2。

我们可以采用第6.3.1节介绍的变分推断方法近似隐变量的联合后验分布。具体来说，我们假设可以用独立的每个隐变量的后验分布之积来近似联合后验分布，即

$$p(\mathbf{h}|\mathbf{x}) \approx q(\mathbf{h}|\mathbf{x}, \phi) = \prod_{i=1}^{m} q(h_i|\mathbf{x}, \phi_i) \tag{10.8}$$

由于隐变量均为二值变量，独立隐变量条件分布可以取伯努利分布

$$q(h_i|\mathbf{x}, \phi_i) = \phi_i^{h_i}(1-\phi_i)^{1-h_i}$$

回顾第6.3.1节，变分推断应求出使得 ELBO 最大的 ϕ。写出 ELBO 表达式，有

$$\begin{aligned}
\text{ELBO}(\phi) &= \langle \log p(\mathbf{x}, \mathbf{z}) \rangle_{q(\mathbf{h}|\mathbf{x},\phi)} - \text{H}[q(\mathbf{z}|\mathbf{x}, \phi)] \\
&= \sum_{\mathbf{h}} (-\log Z + \mathbf{h}^T W\mathbf{x} + \frac{1}{2}\mathbf{h}^T J\mathbf{h} + \frac{1}{2}\mathbf{x}^T K\mathbf{x}) q(\mathbf{h}|\mathbf{x}, \phi) + \text{H}[q(\mathbf{h}|\mathbf{x}, \phi)] \\
&= -\log Z + \frac{1}{2}\mathbf{x}^T K\mathbf{x} + \sum_{\mathbf{h}} (\mathbf{h}^T W\mathbf{x} + \frac{1}{2}\mathbf{h}^T J\mathbf{h}) q(\mathbf{h}|\mathbf{x}, \phi) - \sum_{i=1}^{m} (\phi_i \log \phi_i + (1-\phi_i)\log(1-\phi_i))
\end{aligned}$$

前两项和 ϕ 无关，因此可以写出 ϕ 的损失函数

$$L(\phi) = \sum_{\mathbf{h}} (\mathbf{h}^T W\mathbf{x} + \frac{1}{2}\mathbf{h}^T J\mathbf{h}) q(\mathbf{h}|\mathbf{x}, \phi) - \sum_{i=1}^{m} (\phi_i \log \phi_i + (1-\phi_i)\log(1-\phi_i))$$

其中

$$\begin{aligned}
&\sum_{\mathbf{h}} \mathbf{h}^T W\mathbf{x} q(\mathbf{h}|\mathbf{x}, \phi) \\
=&\sum_{\mathbf{h}} \sum_{i=1}^{m} h_i W_{i,:}\mathbf{x} q(\mathbf{h}|\mathbf{x}, \phi) = \sum_{i=1}^{m} \sum_{\mathbf{h}} h_i W_{i,:}\mathbf{x} q(\mathbf{h}|\mathbf{x}, \phi) \\
=&\sum_{i=1}^{m} \sum_{h_i} \sum_{\mathbf{h}_{-i}} h_i W_{i,:}\mathbf{x} q(\mathbf{h}_{-i}|\mathbf{x}, \phi) q(h_i|\mathbf{x}, \phi) \\
=&\sum_{i=1}^{m} \sum_{h_i} h_i W_{i,:}\mathbf{x} q(h_i|\mathbf{x}, \phi) = \sum_{i=1}^{m} W_{i,:}\mathbf{x} q(h_i = 1|\mathbf{x}, \phi) \\
=&\sum_{i=1}^{m} W_{i,:}\mathbf{x}\phi_i = \phi^T W\mathbf{x}
\end{aligned}$$

$$
\begin{aligned}
&\sum_{\mathbf{h}} \mathbf{h}^T J \mathbf{h} q(\mathbf{h}|\mathbf{x}, \phi) \\
=&\sum_{\mathbf{h}} \sum_{i=1}^{m} \sum_{j=1}^{m} h_i h_j j_{i,j} q(\mathbf{h}|\mathbf{x}, \phi) = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{\mathbf{h}} h_i h_j j_{i,j} q(\mathbf{h}|\mathbf{x}, \phi) \\
=&\sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{h_i, h_j} \sum_{\mathbf{h}_{-i-j}} h_i h_j j_{i,j} q(\mathbf{h}|\mathbf{x}, \phi) \\
=&\sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{h_i, h_j} h_i h_j j_{i,j} q(h_i|\mathbf{x}, \phi_i) q(h_j|\mathbf{x}, \phi_j) \\
=&\sum_{i=1}^{m} \sum_{j=1}^{m} j_{i,j} q(h_i = 1|\mathbf{x}, \phi_i) q(h_j = 1|\mathbf{x}, \phi_j) \\
=&\sum_{i=1}^{m} \sum_{j=1}^{m} \phi_i \phi_j j_{i,j} = \phi^T J \phi
\end{aligned}
$$

因此

$$
L(\phi) = \phi^T W \mathbf{x} + \frac{1}{2} \phi^T J \phi - \sum_{i=1}^{m} (\phi_i \log \phi_i + (1 - \phi_i) \log(1 - \phi_i))
$$

求偏导，有

$$
\frac{\partial L}{\partial \phi_k} = \sum_{i=0}^{n} w_{k,i} x_i + \sum_{j=0}^{m} j_{k,j} \phi_j - \log \frac{\phi_k}{1 - \phi_k}
$$

令偏导为 0，有

$$
\phi_k = \text{sig}(\sum_{i=0}^{n} w_{k,i} x_i + \sum_{j=0}^{m} j_{k,j} \phi_j) \tag{10.9}
$$

于是我们得到了循环依赖的参数更新表达式。只需轮流更新，即可得到隐变量近似后验分布。

10.3 受限玻尔兹曼机 RBM

受限玻尔兹曼机是玻尔兹曼机的简化版本，取消了所有隐变量间的连接和观测变量间的连接。它也是一种生成式模型，常用于数据的无监督特征提取。

具体来说，RBM 假设隐节点 $\mathbf{H} \in \{0, 1\}^m$ 之间互不连接，观测节点 $\mathbf{X} \in \{0, 1\}^n$ 间也互不连接，但隐节点和观测节点有全连接关系，如图10.3所示。

定义能量函数和势函数

$$
E(\mathbf{X}, \mathbf{H}) = -\mathbf{H}^T W \mathbf{X} - \mathbf{a}^T \mathbf{X} - \mathbf{b}^T \mathbf{H} \tag{10.10}
$$

$$
\phi(\mathbf{X}, \mathbf{H}) = \exp\{-E(\mathbf{X}, \mathbf{H})\} \tag{10.11}
$$

其中 $W \in R^{m \times n}, \mathbf{a} \in R^n, \mathbf{b} \in R^m$ 是模型参数。则有联合概率分布

$$
p(\mathbf{X}, \mathbf{H}) = \frac{1}{Z} \exp\{\mathbf{H}^T W \mathbf{X} + \mathbf{a}^T \mathbf{X} + \mathbf{b}^T \mathbf{H}\} \tag{10.12}
$$

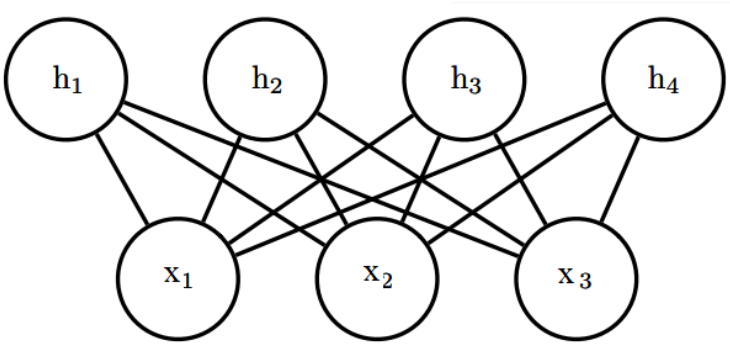

图 10.3: 受限玻尔兹曼机结构示意（图源：花书 16.7.1 节）

根据无向图局部 Markov 性，我们有各隐变量间的条件独立性

$$p(\mathbf{H}|\mathbf{X}) = \prod_{i=1}^{m} p(H_i|\mathbf{X}) \tag{10.13}$$

这一性质是玻尔兹曼机不具有的。

10.3.1 RBM 梯度上升

算法	**受限玻尔兹曼机-梯度上升**
算法简述	对于受限玻尔兹曼机的学习问题，进行最大似然梯度上升的参数迭代求解
已知	样本 $\mathbf{x}^{(i)} \in R^{n\times N}, i = 1, 2, ..., N$，RBM 网络结构，学习率 α
求	参数 $W, \mathbf{a}, \mathbf{b}$ 的最大似然解
解类型	迭代解
算法步骤	1. 随机初始化参数 $W, \mathbf{a}, \mathbf{b}$ 2. 对每一样本 $\mathbf{x}^{(k)}$ 在当前参数 $\theta(t)$ 下使用 Gibbs 采样至少 S 次， 得到 $\mathbf{x}^{(k)}(s)$ 和 $\mathbf{h}^{(k)}(s)$ 3. 按下式计算各参数梯度 $\frac{\partial L}{\partial w_{i,j}} = \frac{1}{N}\sum_{k=1}^{N}\left((x_j^{(k)}(s)\,\mathrm{sig}(W_i\mathbf{x}^{(k)}(s) + b_i h_i^{(k)}(s))) - (x_j^{(k)}\,\mathrm{sig}(W_i\mathbf{x}^{(k)} + b_i h_i^{(k)}(0)))\right)$ $\frac{\partial L}{\partial a_i} = \frac{1}{N}\sum_{k=1}^{N}\left(x_i^{(k)}(s) - x_i^{(k)}\right)$ $\frac{\partial L}{\partial b_i} = \frac{1}{N}\sum_{k=1}^{N}\left(\mathrm{sig}(W_i\mathbf{x}^{(k)}(s) + b_i h_i^{(k)}(s)) - \mathrm{sig}(W_i\mathbf{x}^{(k)} + b_i h_i^{(k)}(0))\right)$ 3. 进行梯度上升参数更新 $\theta(t+1) = \theta(t) + \alpha\frac{\partial L}{\partial \theta}$ 4. 若似然函数收敛，停止迭代，否则返回 2。

RBM 的参数学习仍然需要求解最大似然。我们可以直接使用式10.7的结论

$$\frac{\partial L}{\partial \theta} = -\left\langle \frac{\partial L}{\partial \theta} E(\mathbf{x}, \mathbf{h}) \right\rangle_{p(\mathbf{h}|\mathbf{x})} + \left\langle \left\langle \frac{\partial L}{\partial \theta} E(\mathbf{X}, \mathbf{H}) \right\rangle_{p(\mathbf{H}|\mathbf{X})} \right\rangle_{p(\mathbf{X})}$$

具体到三个参数 $W, \mathbf{a}, \mathbf{b}$，有

$$\begin{aligned}
\left\langle \frac{\partial}{\partial w_{i,j}} E(\mathbf{x},\mathbf{h}) \right\rangle_{p(\mathbf{h}|\mathbf{x})} &= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) \frac{\partial}{\partial w_{i,j}} E(\mathbf{x},\mathbf{h}) \\
&= \sum_{\mathbf{h}} \prod_{k=1}^{M} p(H_k|\mathbf{x}) H_i x_j = \sum_{H_i} p(H_i|\mathbf{x}) \sum_{H_{-i}} p(H_{-i}|\mathbf{x}) H_i x_j \\
&= \sum_{H_i} p(H_i|\mathbf{x}) H_i x_j = p(H_i=0|\mathbf{x}) 0 x_j + p(H_i=1|\mathbf{x}) 1 x_j \\
&= p(H_i=1|\mathbf{x}) x_j \\
\left\langle \frac{\partial}{\partial b_i} E(\mathbf{x},\mathbf{h}) \right\rangle_{p(\mathbf{h}|\mathbf{x})} &= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) \frac{\partial}{\partial b_i} E(\mathbf{x},\mathbf{h}) \\
&= \sum_{\mathbf{h}} \prod_{k=1}^{M} p(H_k|\mathbf{x}) H_i = \sum_{H_i} p(H_i|\mathbf{x}) \sum_{H_{-i}} p(H_{-i}|\mathbf{x}) H_i \\
&= \sum_{H_i} p(H_i|\mathbf{x}) H_i = p(H_i=0|\mathbf{x}) 0 + p(H_i=1|\mathbf{x}) 1 \\
&= p(H_i=1|\mathbf{x}) \\
\left\langle \frac{\partial}{\partial a_i} E(\mathbf{x},\mathbf{h}) \right\rangle_{p(\mathbf{h}|\mathbf{x})} &= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) \frac{\partial}{\partial a_i} E(\mathbf{x},\mathbf{h}) \\
&= \sum_{\mathbf{h}} \prod_{k=1}^{M} p(H_k|\mathbf{x}) x_i = x_i
\end{aligned}$$

由条件独立性，我们可以直接计算单隐变量在观测变量条件下的分布

$$\begin{aligned}
p(h_i=1|\mathbf{x}) &= p(h_i=1|\mathbf{h}_{-i},\mathbf{x}) \\
&= \frac{p(h_i=1,\mathbf{h}_{-i},\mathbf{x})}{p(h_i=1,\mathbf{h}_{-i},\mathbf{x}) + p(h_i=0,\mathbf{h}_{-i},\mathbf{x})}
\end{aligned}$$

记 $W,\mathbf{b}$ 去掉 h_i 对应元素后的矩阵和向量为 $W_{-i},\mathbf{b}_{-i}$，记

$$\phi(\mathbf{h}_{-i},\mathbf{x}) = \exp\{\mathbf{h}_{-i}^T W_{-i}\mathbf{x} + \mathbf{a}^T\mathbf{x} + \mathbf{b}_{-i}^T\mathbf{h}_{-i}\}$$

则有

$$\begin{aligned}
p(h_i=1|\mathbf{x}) &= p(h_i=1|\mathbf{h}_{-i},\mathbf{x}) \\
&= \frac{p(h_i=1,\mathbf{h}_{-i},\mathbf{x})}{p(h_i=1,\mathbf{h}_{-i},\mathbf{x}) + p(h_i=0,\mathbf{h}_{-i},\mathbf{x})} \\
&= \frac{\frac{1}{Z}\phi(\mathbf{h}_{-i},\mathbf{x})\exp\{W_i\mathbf{x}+b_ih_i\}}{\frac{1}{Z}\phi(\mathbf{h}_{-i},\mathbf{x})\exp\{W_i\mathbf{x}+b_ih_i\} + \frac{1}{Z}\phi(\mathbf{h}_{-i},\mathbf{x})} \\
&= \frac{\exp\{W_i\mathbf{x}+b_ih_i\}}{\exp\{W_i\mathbf{x}+b_ih_i\}+1} \\
&= \mathrm{sig}(W_i\mathbf{x}+b_ih_i)
\end{aligned}$$

因此我们有梯度表达式

$$\begin{aligned}\frac{\partial L}{\partial w_{i,j}} =& - x_j \operatorname{sig}(W_i\mathbf{x} + b_i h_i) + \langle X_j \operatorname{sig}(W_i\mathbf{X} + b_i H_i)\rangle_{p(\mathbf{X})} \\ \frac{\partial L}{\partial a_i} =& - x_i + \langle X_i \rangle_{p(\mathbf{X})} \\ \frac{\partial L}{\partial b_i} =& - \operatorname{sig}(W_i\mathbf{x} + b_i h_i) + \langle \operatorname{sig}(W_i\mathbf{X} + b_i H_i)\rangle_{p(\mathbf{X})}\end{aligned}$$

考虑对样本集所有样本梯度求平均，我们有

$$\begin{aligned}\frac{\partial L}{\partial w_{i,j}} =& - \frac{1}{N}\sum_{k=1}^{N} x_j^{(k)} \operatorname{sig}(W_i\mathbf{x}^{(k)} + b_i h_i^{(k)}) + \langle X_j \operatorname{sig}(W_i\mathbf{X} + b_i H_i)\rangle_{p(\mathbf{X})} \\ \frac{\partial L}{\partial a_i} =& - \frac{1}{N}\sum_{k=1}^{N} x_i^{(k)} + \langle X_i \rangle_{p(\mathbf{X})} \\ \frac{\partial L}{\partial b_i} =& - \frac{1}{N}\sum_{k=1}^{N} \operatorname{sig}(W_i\mathbf{x}^{(k)} + b_i h_i^{(k)}) + \langle \operatorname{sig}(W_i\mathbf{X} + b_i H_i)\rangle_{p(\mathbf{X})}\end{aligned}$$

注意，和玻尔兹曼机相对比，RBM 梯度的负相部分可以直接求出解析解。这是 RBM 的结构限制带来的条件独立性造成的便利。不过其中的 $\mathbf{h}^{(k)}$ 是对应于样本 $\mathbf{x}^{(k)}$ 的隐变量。这个变量我们无法得知，只能从 RBM 中采样得到。

另一方面，正相仍然无法解析求解。我们仍然使用第6章介绍的 Gibbs 采样方法。但由于条件独立性，我们在采样隐变量时无需依赖其他隐变量，同样对于观测变量的采样也不需要其他观测变量。因此，我们将 RBM 的变量按隐变量和观测变量分为两组，轮流进行采样。

这样，我们就可以计算梯度的两个部分。计算负相时，以当前样本 $\mathbf{x}^{(k)}$ 为观测变量，采样隐变量 $\mathbf{h}^{(k)}$，相当于 Gibbs 采样的第 0 轮。计算正相时，以当前样本集的样本为 Gibbs 采样的初始样本，记样本 $\mathbf{x}^{(k)}$ 经过 s 次 Gibbs 采样后的样本为 $\mathbf{x}^{(k)}(s)$，则有

$$\langle F(\mathbf{X})\rangle_{p(\mathbf{X})} = \frac{1}{N}\sum_{k=1}^{N} F(\mathbf{x}^{(k)}(s)), s > S$$

其中 S 为状态分布收敛到平稳分布的最小步数。因此最终的梯度表达式可以写为

$$\begin{aligned}\frac{\partial L}{\partial w_{i,j}} =& \frac{1}{N}\sum_{k=1}^{N} \left((x_j^{(k)}(s) \operatorname{sig}(W_i\mathbf{x}^{(k)}(s) + b_i H_i^{(k)}(s))) - (x_j^{(k)} \operatorname{sig}(W_i\mathbf{x}^{(k)} + b_i h_i^{(k)})) \right) \\ \frac{\partial L}{\partial a_i} =& \frac{1}{N}\sum_{k=1}^{N} \left(x_i^{(k)}(s) - x_i^{(k)} \right) \\ \frac{\partial L}{\partial b_i} =& \frac{1}{N}\sum_{k=1}^{N} \left(\operatorname{sig}(W_i\mathbf{x}^{(k)}(s) + b_i H_i^{(k)}(s)) - \operatorname{sig}(W_i\mathbf{x}^{(k)} + b_i h_i^{(k)}) \right)\end{aligned} \tag{10.14}$$

10.3.2 对比散度算法

算法	**受限玻尔兹曼机-对比散度**
算法简述	对于受限玻尔兹曼机的学习问题，进行最大似然参数的梯度计算迭代求解
已知	样本 $\mathbf{x}^{(i)} \in R^{n\times N}, i = 1, 2, ..., N$，RBM 网络结构，学习率 α
求	参数 $W, \mathbf{a}, \mathbf{b}$ 的最大似然解
解类型	迭代解
算法步骤	1. 随机初始化参数 $\theta = W, \mathbf{a}, \mathbf{b}$ 2. 对每一样本 $\mathbf{x}^{(k)}$ 在当前参数 $\theta(t)$ 下使用 Gibbs 采样得到 $\mathbf{h}^{(k)}$， 迭代一次得到 $\mathbf{x}^{(k)}(1)$ 和 $\mathbf{h}^{(k)}(1)$ 3. 按下式计算各参数梯度 $\frac{\partial L}{\partial w_{i,j}} = \frac{1}{N}\sum_{k=1}^{N}\left((x_j^{(k)}(1)\,\mathrm{sig}(W_i\mathbf{x}^{(k)}(1) + b_i h_i^{(k)}(1))) - (x_j^{(k)}\,\mathrm{sig}(W_i\mathbf{x}^{(k)} + b_i h_i^{(k)}))\right)$ $\frac{\partial L}{\partial a_i} = \frac{1}{N}\sum_{k=1}^{N}\left(x_i^{(k)}(1) - x_i^{(k)}\right)$ $\frac{\partial L}{\partial b_i} = \frac{1}{N}\sum_{k=1}^{N}\left(\mathrm{sig}(W_i\mathbf{x}^{(k)}(1) + b_i h_i^{(k)}(1)) - \mathrm{sig}(W_i\mathbf{x}^{(k)} + b_i h_i^{(k)})\right)$ 3. 进行梯度上升参数更新 $\theta(t+1) = \theta(t) + \alpha\frac{\partial L}{\partial \theta}$ 4. 若似然函数收敛，停止迭代，否则返回 2。

回顾第6章中，我们提到 Gibbs 采样的核心是构造平稳分布为目标分布的马尔可夫链。为此，我们需要等待一定采样步数直到状态分布收敛到平稳分布。然而，梯度上升法中，每一步梯度的更新都需要等待收敛，这无疑带来巨大的时间成本。并且，随着节点规模增加，等待时间 S 将快速增长，直到采样事实上不可计算。

假设在对参数 θ 更新的第 t 个时间步，$\mathbf{x}$ 的边缘分布为 $p(\mathbf{X}|\theta(t))$，Gibbs 采样的第 s 个时间步的状态分布为 $\mathbf{x}(s) \sim \pi(\mathbf{x}|s, \theta(t))$，理论上有

$$\pi(\mathbf{x}|S+1, \theta(t)) = \pi(\mathbf{x}|S+2, \theta(t)) = ... = \pi(\mathbf{x}|\infty, \theta(t)) = p(\mathbf{x}|\theta(t)) \tag{10.15}$$

梯度下降算法的梯度10.14可以写为

$$\frac{\partial L}{\partial \theta} = G(\mathbf{x}(s), \theta(t)) - G(\mathbf{x}(0), \theta(t))$$

其中

$$G(\mathbf{x}, \theta) = \left\langle \frac{\partial L}{\partial \theta} E(\mathbf{x}, \mathbf{h}|\theta) \right\rangle_{p(\mathbf{h}|\mathbf{x},\theta)}$$

Gibbs 算法的每个时间步，可以看作是在优化当前时间步分布和目标分布间的 KL 散度

$$\min_{\pi} \mathrm{KL}\left[\pi(\mathbf{x}|s, \theta(t))||\pi(\mathbf{x}|\infty, \theta(t))\right]$$

对于参数 θ 的每步更新，如果都要等待至少 S 步，显然是非常耗时的。然而，根据式10.15，$\pi(\mathbf{x}|S+1, \theta(t)) = \pi(\mathbf{x}|\infty, \theta(t))$ 时也有 $\pi(\mathbf{x}|S+1, \theta(t)) = \pi(\mathbf{x}|S+2, \theta(t))$。因此，假设优化下式也能起到类似的效果

$$\min_{\pi} \mathrm{KL}\left[\pi(\mathbf{x}|s, \theta(t))||\pi(\mathbf{x}|s+1, \theta(t))\right]$$

从而，Gibbs 采样不需要迭代 S 步而只需迭代 1 步，得到的梯度也可能有效

$$\frac{\partial L}{\partial \theta} = G(\mathbf{x}(1), \theta(t)) - G(\mathbf{x}(0), \theta(t)) \tag{10.16}$$

这样就可以大幅减少 Gibbs 采样的等待时间，形成一种高效的梯度下降算法。由于这个算法的来源是对比了迭代一次后的分布散度，因此也称为对比散度算法。该算法没有严谨的有效性证明，但各类实验结果都说明它是一种高效且有效的训练算法）

10.3.3 深度化模型

RBM 在训练完成后，可以作为高维数据的特征提取器，或起到降维的效果，可以用于无监督任务或者有监督任务。从结构上来看，它相当于一种单层神经网络。通过堆叠 RBM，可以形成深层神经网络，也就是深度信念网络 (DBN) 和深度玻尔兹曼机 (DBM)（图10.4）。

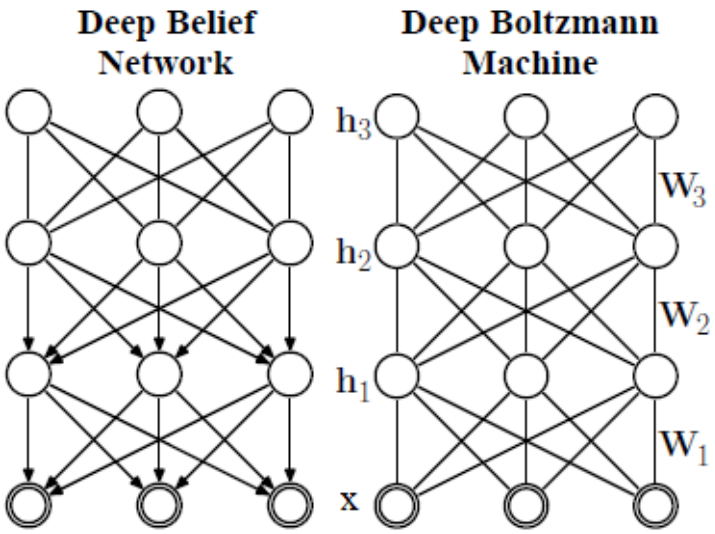

图 10.4: DBN（左）与 DBM（右）

深度信念网络从结构上是一层 RBM 和多层的 SBN 的结合，但它的参数是由下而上逐层贪心训练 RBM 得到的。它可以作为一种混合了有向图和无向图的生成模型参与监督学习或非监督学习任务。深度玻尔兹曼机结构上就是多层 RBM 的结合。它的训练过程比 DBN 更加复杂，此处不再赘述。

11 时序概率图模型

本章将介绍一系列表征时序数据的概率图模型。在深度学习时代到来之前，它们就已经是自然语言处理领域的重要工具。为简单起见，本章介绍的所有时序概率图都是线性链结构。

11.1 基本任务

对于含有隐变量的时序概率图模型，我们会有推断一些特定形式的概率分布的需求。此外，由于特殊的图结构，一些对于此前介绍的概率图模型来说比较简单的分布也不容易求得。因此，产生了一系列时序隐变量概率图模型相关的任务。

模型评估任务

模型评估任务，就是在给定模型参数 θ 的条件下，对于已知的观测序列（或样本）$\mathbf{x}$，计算其边缘概率分布 $p(\mathbf{x}|\theta)$。

对于先前介绍的概率图模型，如 SBN、RBM 等，所有观测变量的联合边缘概率分布较容易解析表达，也可以推导出较为简便的计算方法。但对于时序概率图来说，它的序列结构使得直接计算比较困难。

隐变量推断任务

隐变量推断任务，也称为**解码任务**，是在给定模型参数 θ 的条件下，对于已知的全部观测序列（或样本）$\mathbf{x}$，计算后验概率最大的隐变量序列 $\hat{\mathbf{z}}$，即 $\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta)$。

参数学习任务

和此前介绍的一样，参数学习任务是在已知样本集 X、模型结构的条件下，求出使似然函数 $p(\mathbf{x}|\theta)$ 最大的一组参数 $\hat{\theta}$。

如果改变已知和未知的时间条件，隐变量推断任务还可以衍生出下列几个任务：

滤波任务

滤波任务，是在给定从开始到当前时刻观测序列 $x_1, x_2, ..., x_t$ 的条件下，计算当前时刻隐变量的概率分布 $p(z_t|x_1, x_2, ..., x_t)$。

平滑任务

平滑任务，是在给所有时刻观测序列 $x_1, x_2, ..., x_T$ 的条件下，计算某一时刻隐变量的概率分布 $p(z_t|x_1, x_2, ..., x_T)$。

预测任务

预测任务，是在给定从开始到当前时刻观测序列 $x_1, x_2, ..., x_t$ 的条件下，计算未来时刻隐变量的概率分布 $p(z_{t+1}, z_{t+2}, ..., z_T|x_1, x_2, ..., x_t)$。

11.2 隐马尔可夫模型 HMM

隐马尔可夫模型是一种含隐变量的有向概率图模型，用于处理时序数据。

11.2.1 基本概念

由于隐马尔可夫模型涉及随时间变化的含隐变量概率图模型，需要重新规范一套符号约定。本节和下一节的所有符号均遵守下列符号约定。

假设有（随时间变化的）离散型随机变量序列 $\mathbf{X} = \{X_1, X_2, ..., X_T\}$。其中 X_t 的所有可能取值集合为 $\mathbf{V} = \{v_1, v_2, ..., v_V\}$，称为**观测集**。相应的随机变量序列 $\{X_t\}$ 称为**观测序列**。

隐马尔可夫模型假设随机变量 X_t 仅由同一时刻的隐变量 Z_t 控制（**齐次马尔可夫假设**），且隐变量 Z_t 仅由上一时刻的隐变量 Z_{t-1} 控制（**观测独立假设**）[3]。此处的隐变量序列 $\mathbf{Z} = \{Z_1, Z_2, ..., Z_T\}$ 也称为**状态序列**。隐马尔可夫模型假设隐变量也是离散型随机变量，其取值范围为 $\mathbf{S} = \{s_1, s_2, ..., s_S\}$，称为**状态集**。

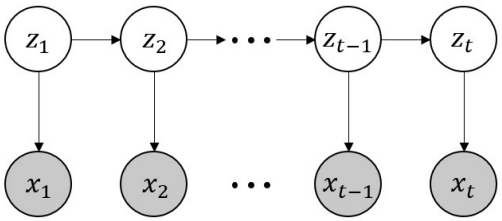

图 11.1: 隐马尔可夫模型结构

根据上述假设，存在两个条件分布 $p(X_t|Z_t)$ 和 $p(Z_t|Z_{t-1})$。这两个条件分布分别称为**状态转移分布**和**观测分布**。由于隐变量和观测变量均为离散型随机变量，状态转移分布和观测分布分别可以表示成矩阵形式，即**状态转移矩阵** $A \in R^{S\times S}$ 和**观测概率矩阵** $B = R^{S\times V}$。即

$$p(Z_t = s_j|Z_{t-1} = s_i) = p(s_j|s_i) = a_{ij} \tag{11.1}$$

$$p(X_t = v_j|Z_t = s_i) = p(v_i|s_j) = b_{ij} \tag{11.2}$$

其中

$$\sum_{j=1}^{S} a_{ij} = 1$$

$$\sum_{j=1}^{V} b_{ij} = 1$$

为书写方便，定义发射函数

$$b_i(x_t) = p(X_t = x_t|Z_t = s_i) \tag{11.3}$$

此外，假设初始状态 Z_1 服从一个多元伯努利概率分布，其参数为 π，即

$$p(Z_1 = s_i) = \pi_i \tag{11.4}$$

其中

$$\sum_{i=1}^{S} \pi_i = 1$$

为书写方便，定义初始分布函数

$$\pi(s_i) = p(Z_1 = s_i)$$

一个隐马尔可夫模型就是由这样的 (A, B, π) 三元组所定义的。其图模型结构如图11.1所示。

[3]事实上也有不满足这一假设的 HMM 模型，根据条件概率考虑的此前隐变量的多少称为 k 阶马尔可夫模型。本书只介绍一阶马尔可夫模型。

11.2.2 基本任务

HMM 需要处理的基本任务就是上节介绍的模型评估、隐变量推断和参数学习任务。

对于模型评估任务，HMM 的隐变量共有 T 个，每个隐变量取值有 S 种可能性，因此一共有 S^T 种隐变量取值组合。在已知隐变量条件下观测变量相互独立，对于每种组合需要进行 T 次乘法算出联合分布 $p(\mathbf{x},\mathbf{z})$。因此暴力算法的总计算复杂度为 $O(TS^T)$。这是一个状态集规模的指数的复杂度，显然需要一些更简单的算法来计算。

对于 HMM 而言，隐变量后验只能使用贝叶斯公式求解

$$p(\mathbf{Z}|\mathbf{X},A,B,\pi)=\frac{p(\mathbf{Z},\mathbf{X}|A,B,\pi)}{p(\mathbf{X}|A,B,\pi)}$$

前面计算过，对于联合分布 $p(\mathbf{Z},\mathbf{X}|A,B,\pi)$ 的每一种取值 $p(\mathbf{x},\mathbf{z})$，都需要 T 次乘法。全部取值可能性为 S^T。同时分母也需要计算全部 S^T 种可能，因此隐变量推断的暴力算法总计算复杂度也为 $O(TS^T)$。

和其他概率图模型一样，隐马尔可夫模型也有模型参数的学习任务。具体来说，给定包含观测序列的数据集 Dataset $=\{\mathbf{x}^{(1)},\mathbf{x}^{(2)},...,\mathbf{x}^{(N)}\}$，求使得似然（即观测变量联合边缘分布）最大的参数，称为**无监督参数学习**。给定包含观测序列和对应状态序列的数据集 Dataset $=\{\mathbf{x}^{(1)},\mathbf{x}^{(2)},...,\mathbf{x}^{(N)},\mathbf{z}^{(1)},\mathbf{z}^{(2)},...,\mathbf{z}^{(N)}\}$，求使得似然（即观测变量和隐变量联合分布）最大的参数，称为**有监督参数学习**。

对于有监督参数学习，根据上文计算，计算一组参数下的联合分布 $p(\mathbf{x}^{(i)},\mathbf{z}^{(i)}|A,B,\pi)$ 的复杂度为 $O(NT)$。如果使用迭代的梯度下降算法，需要每轮迭代都计算联合分布，并且对各参数进行反向传播。对于无监督参数学习，每轮迭代就要计算边缘分布 $p(\mathbf{x}^{(i)}|A,B,\pi)$，暴力计算复杂度为 $O(NTS^T)$。这显然是不可接受的。

因此，接下来的几小节中，我们将针对 HMM 的特点，构造低计算复杂度的基本任务算法。为简单起见，对于评估问题和隐变量推断问题，我们往往在概率表达中省略以参数为条件。

11.2.3 评估算法

算法	**隐马尔可夫-前向算法**
算法简述	对于隐马尔可夫模型的评估问题，使用前向概率函数迭代计算
已知	样本 $\mathbf{x}=\{x_1,x_2,...,x_T\}$，状态转移矩阵 $A\in R^{S\times S}$， 观测概率矩阵 $B\in R^{S\times V}$，初始分布 $\pi\in R^S$
求	边缘概率 $p(\mathbf{x}\|A,B,\pi)$
解类型	闭式解
算法步骤	1. 计算初始前向概率 $\alpha_1(i)=b_i(x_1)\pi_i$，取 $t=1$ 2. 计算前向概率 $\alpha_t(i)=b_i(x_t)\sum_{j=1}^S\alpha_{t-1}(j)b_{ji}$ 3. 若 $t<T$，令 $t\leftarrow t+1$，返回 2。否则计算边缘概率 $p(\mathbf{X})=\sum_{j=1}^S\alpha_T(i)$

对于评估问题，我们先考虑在部分观测序列下末尾隐变量的分布。对于部分观测序列 $x_1,x_2,...,x_t$ 以及 t 时刻的状态 Z_t，定义状态 Z_t 取 s_i 时的联合分布为**前向概率函数**

$$\begin{aligned}\alpha_t(i)=&p(Z_t=s_i,X_1=x_1,X_2=x_2,...,X_t=x_t)\\=&p(Z_t=s_i,x_1,x_2,...,x_t),\ t=1,2,...,T\end{aligned}\tag{11.5}$$

这一定义类似于滤波任务的结果，只不过要求的是联合分布。由此可以推导出前向概率函数的递推公式

$$
\begin{aligned}
\alpha_t(i) =& p(s_i, x_1, x_2, ..., x_t) \\
=& p(x_t|s_i)p(Z_t = s_i, x_1, x_2, ..., x_{t-1}) \\
=& p(x_t|s_i)p(Z_{t-1}, x_1, x_2, ..., x_{t-1})p(Z_t = s_i|Z_{t-1}) \\
=& p(x_t|s_i)\sum_{j=1}^{S} p(Z_{t-1} = s_j, x_1, x_2, ..., x_{t-1})p(Z_t = s_i|Z_{t-1} = s_j) \\
=& b_i(x_t)\sum_{j=1}^{S} \alpha_{t-1}(j)b_{ji}
\end{aligned}
$$

特别地，我们有

$$\alpha_1(i) = p(x_1|s_i)p(Z_1 = s_i) = b_i(x_1)\pi_i$$

因此，我们可以迭代求解除任意时刻的前向概率函数。注意到观测序列边缘分布 $p(\mathbf{X})$ 和 $t = T$ 时前向概率函数的关系

$$p(\mathbf{x}) = \sum_{i=1}^{S} p(Z_T = s_i, x_1, x_2, ..., x_t) = \sum_{j=1}^{S} \alpha_T(i)$$

因此我们可以迭代地计算出 $p(\mathbf{x})$：

$$
\begin{aligned}
\alpha_1(i) =& b_i(x_1)\pi_i \\
\alpha_t(i) =& b_i(x_t)\sum_{j=1}^{S} \alpha_{t-1}(j)b_{ji} \\
p(\mathbf{x}) =& \sum_{j=1}^{S} \alpha_T(i)
\end{aligned}
\tag{11.6}
$$

这一算法也被称为**前向算法**。在前向算法中，对于每一个 $\alpha_t(i)$，乘法和加法的次数都仅仅是 $O(S)$。因此对于每个时间步 t，计算全部 $\alpha_t(i)$ 的计算复杂度为 $O(S^2)$。迭代计算 $p(\mathbf{x})$ 的时间复杂度仅为 $O(TS^2)$，明显优于暴力算法的 $O(TS^T)$。

算法	**隐马尔可夫-后向算法**
算法简述	对于隐马尔可夫模型的评估问题，使用后向概率函数迭代计算
已知	样本 $\mathbf{x} = \{x_1, x_2, ..., x_T\}$，状态转移矩阵 $A \in R^{S\times S}$， 观测概率矩阵 $B \in R^{S\times V}$，初始分布 $\pi \in R^S$
求	边缘概率 $p(\mathbf{x}|A, B, \pi)$
解类型	闭式解
算法步骤	1. 计算初始后向概率 $\beta_T(i) = 1$，取 $t = T$ 2. 计算后向概率 $\beta_t(i) = \sum_{j=1}^{S} a_{ij}b_j(x_{t+1})\beta_{t+1}(j)$ 3. 若 $t > 1$，令 $t \leftarrow t - 1$，返回 2。否则计算边缘概率 $p(\mathbf{x}) = \sum_{i=1}^{S} \pi_i b_i(x_1)\beta_1(i)$

类似地，我们也可以定义后向概率函数

$$\begin{aligned}\beta_t(i) =& p(X_{t+1}=x_{t+1}, X_{t+2}=x_{t+2}, ..., X_T=x_T|Z_t=s_i)\\ =& p(x_{t+1}, x_{t+2}, ..., x_T|Z_t=s_i),\ t=1,2,...,T\end{aligned} \tag{11.7}$$

注意和前向概率函数不同，这里的隐变量是条件。这样，我们有后向迭代式

$$\begin{aligned}\beta_t(i) =& p(x_{t+1}, x_{t+2}, ..., x_T|Z_t=s_i)\\ =& \sum_{j=1}^{S} p(Z_{t+1}=s_j, x_{t+1}, x_{t+2}, ..., x_T|Z_t=s_i)\\ =& \sum_{j=1}^{S} p(Z_{t+1}=s_j|Z_t=s_i)p(x_{t+1}, x_{t+2}, ..., x_T|Z_{t+1}=s_j, Z_t=s_i)\\ =& \sum_{j=1}^{S} p(Z_{t+1}=s_j|Z_t=s_i)p(x_{t+1}|Z_{t+1}=s_j)p(x_{t+2}, ..., x_T|Z_{t+1}=s_j)\\ =& \sum_{j=1}^{S} a_{ij}b_j(x_{t+1})\beta_{t+1}(j)\end{aligned}$$

特别地，$t=T$ 时的后向概率

$$\beta_T(i)=1$$

注意到观测序列边缘分布 $p(\mathbf{x})$ 和 $t=1$ 时后向概率函数的关系

$$\begin{aligned}p(\mathbf{x}) =& \sum_{i=1}^{S} p(Z_1=s_i)p(x_1, x_2, ..., x_T|Z_1=s_i)\\ =& \sum_{i=1}^{S} p(Z_1=s_i)p(x_1|Z_1=s_i)p(x_2, x_3, ..., x_T|Z_1=s_i)\\ =& \sum_{i=1}^{S} \pi_i b_i(x_1)\beta_1(i)\end{aligned}$$

因此我们可以迭代地计算出 $p(\mathbf{x})$：

$$\begin{aligned}\beta_T(i) =& 1\\ \beta_t(i) =& \sum_{j=1}^{S} a_{ij}b_j(x_{t+1})\beta_{t+1}(j)\\ p(\mathbf{x}) =& \sum_{i=1}^{S} \pi_i b_i(x_1)\beta_1(i)\end{aligned} \tag{11.8}$$

该算法也被称为**后向算法**。在后向算法中，对于每一个 $\beta_t(i)$，乘法的次数仅仅是 $O(S)$。因此对于每个时间步 t，计算全部 $\beta_t(i)$ 的计算复杂度为 $O(S^2)$。迭代计算 $p(\mathbf{x})$ 的时间复杂度也为 $O(TS^2)$，优于暴力算法的 $O(TS^T)$。

11.2.4 隐变量推断算法

动态规划是一种解决多阶段最优决策问题的思想。它基于贝尔曼提出的最优性原理，即：若一个最优决策问题可以分解为多个相同构造的子问题的叠加，那么父问题最优解必定包含子问题的最优解。

使用动态规划算法求解问题的典型例子是带权图最短路径问题，而 HMM 的隐变量推断问题也可以看成一个路径规划问题。具体来说，可以认为每两个时间步之间的隐变量后验分布 $p(Z_t|Z_{t-1}, X_t)$ 对应两层隐变量取值节点之间的全连接图。图的权重即为隐变量不同取值对应的单步后验分布 $p(s_i|s_j, X_t)$。最优后验分布对应的隐变量取值列即为最大权重的路径，即

$$\max_{Z} p(\mathbf{Z}|\mathbf{X}) = \max_{Z_1} p(Z_1|X_1) \prod_{t=2}^{T} \max_{Z_t} p(Z_t|Z_{t-1}, X_t)$$

其中

$$\begin{aligned}\max_{Z_t} p(Z_t|Z_{t-1}, X_t) &= \max_{Z_t} \frac{p(Z_t, X_t|Z_{t-1})}{p(X_t|Z_{t-1})} \\ &= \max_{Z_t} \frac{p(Z_t|Z_{t-1})p(X_t|Z_t)}{p(X_t|Z_{t-1})}\end{aligned}$$

这样，使用逐步贪心的测略，我们就可以得到 HMM 的隐变量推断算法，也称为 **Viterbi 算法**。

算法	**隐马尔可夫-Viterbi 算法**
算法简述	对于隐马尔可夫模型的隐变量推断问题，使用动态规划思想分布求解
已知	样本 $\mathbf{x} = \{x_1, x_2, ..., x_T\}$，状态转移矩阵 $A \in R^{S\times S}$， 观测概率矩阵 $B \in R^{S\times V}$，初始分布 $\pi \in R^S$
求	最优隐变量序列 $\arg\max_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, A, B, \pi)$
解类型	闭式解
算法步骤	1. 求初始最可能序列概率 $\delta_1(i) = \pi_i b_i(x_1)$ 2. 迭代求解最可能序列概率 $\delta_t(i) = \max_j a_{ji} b_i(x_t)\delta_{t-1}(j)$ 同时记录前一状态 $c_t(i) = \arg\max_j a_{ji} b_i(x_t)\delta_{t-1}(j)$ 3. $t \leftarrow t+1$, 若 $t < T$ 回 2，否则到 4 4. 计算最优末状态 $i_T = \arg\max_i \delta_T(i)$ 5. 依次求取最优状态 $i_t = c_t(i_{t+1})$，并且 $\hat{z}_t = s_{i_t}$，$t = 1, 2, ..., T$

具体来说，假设 $\delta_t(i)$ 表示从时刻 1 到时刻 t 的所有可能隐变量序列中，满足 $Z_t = s_i$ 的最大的序列后验概率，即

$$\delta_t(i) = \max_{z_1, z_2, ..., z_{t-1}} p(z_1, z_2, ..., z_{t-1}|z_t = s_i, x_1, x_2, ..., x_t) \tag{11.9}$$

有递推关系式

$$\begin{aligned}\delta_t(i) &= \max_{z_1, z_2, ..., z_{t-1}} p(z_1, z_2, ..., z_{t-1}|z_t = s_i, x_1, x_2, ..., x_t) \\ &= \max_j p(z_t = s_i|z_{t-1} = s_j, x_t) \max_{z_1, z_2, ..., z_{t-2}} p(z_1, z_2, ..., z_{t-2}|z_{t-1} = s_j, x_1, x_2, ..., x_t) \\ &= \max_j p(s_i|s_j)p(x_t|s_i) \max_{z_1, z_2, ..., z_{t-2}} p(z_1, z_2, ..., z_{t-2}|z_{t-1} = s_j, x_1, x_2, ..., x_{t-1}) \\ &= \max_j a_{ji} b_i(x_t)\delta_{t-1}(j)\end{aligned}$$

特别地，有

算法	**隐马尔可夫-Baum-Welch 算法**
算法简述	对于隐马尔可夫模型的参数学习问题，使用 E-M 算法求解
已知	样本 Dataset $= \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$
求	最优参数，包括状态转移矩阵 $A \in R^{S\times S}$、观测概率矩阵 $B \in R^{S\times V}$、 初始分布 $\pi \in R^S$，满足各自约束条件
解类型	迭代解
算法步骤	1. 任意初始化参数 $A(0), B(0), \pi(0)$ 2. (E 步) 在时间步 τ，对每一个样本 $\mathbf{x} \in$ Dataset，使用前向算法和后向算法计算 $\alpha_t(i)(\tau),\ t = 1, 2, ..., T,\ i = 1, 2, ..., S,\ \beta_t(i)(\tau),\ t = 1, 2, ..., T,\ i = 1, 2, ..., S$ 3. (E 步) 计算以下中间结果（省略时间步 τ 标记） $\gamma_{ij} = \sum_{t=2}^{T} \alpha_{t-1}(i) a_{ij} b_j(x_t) \beta_t(j)$ $\eta_{ij} = \sum_{t=2}^{T} \alpha_t(i) \beta_t(j)$ $p(Z_1 = s_i, \mathbf{x}\|\theta_{old}) = \pi(i)\beta_1(i)b_1(x_1)$ $p(\mathbf{x}\|\theta_{old}) = \sum_{i=1}^{S} \pi(i)\beta_1(i)b_1(x_1)$ 4. (E 步) 计算中间结果对所有样本 $\mathbf{x} \in$ Dataset 的平均值 5. (M 步) 按照下列公式更新参数 $a_{ij}(\tau+1) = \frac{\gamma_{ij}(\tau)}{\sum_{k=1}^{S}\gamma_{ik}(\tau)}$ $\pi_i(\tau+1) = \frac{p(Z_1=s_i,\mathbf{x}\|\theta_{old})(\tau)}{p(\mathbf{x}\|\theta_{old})(\tau)}$ $b_{ij}(\tau+1) = \frac{\eta_{ij}(\tau)}{\sum_{k=1}^{V}\eta_{ik}(\tau)}$ 6. $\tau \leftarrow \tau + 1$，若各参数收敛，停止迭代，否则返回 2

$$\delta_1(i) = \pi_i b_i(x_1)$$

其中需要记录每一步取最大时具体取了哪一状态

$$c_t(i) = \arg\max_j \ a_{ji} b_i(x_t) \delta_{t-1}(j)$$

在计算完成后，取

$$i_T = \arg\max_i \ \delta_T(i)$$

且依次倒推

$$i_t = c_t(i_{t+1}),\ t = 1, 2, ..., T-1$$

即可得到最优状态序列

$$\hat{z}_t = s_{i_t},\ t = 1, 2, ..., T$$

11.2.5 参数学习算法

HMM 是一个含有隐变量的概率图模型。对于它的无监督参数学习问题，和此前遇到的很多类似问题一样，我们选用 E-M 算法来解决参数最大似然估计问题。具体来说，我们的目标是优化对数似然（假设只考虑

一个样本 $\mathbf{x}$）

$$\max_{A,B,\pi} \log p(\mathbf{x}|A,B,\pi)$$

下面的推导中，有时为表示方便，我们会把参数 A,B,π 合写为 θ。由于对数似然求解非常复杂，我们考虑优化熵函数（6.3.2节的式6.20）

$$HF(\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}=\mathbf{x},\mathbf{Z}|\theta_{old}) \log p(\mathbf{X}=\mathbf{x},\mathbf{Z}|\theta)$$

由于

$$\begin{aligned}
\log p(\mathbf{x},\mathbf{z}|\theta) &= \log \prod_{t=1}^{T} p(x_t|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1},\theta) p(z_t|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1},\theta) \\
&= \log(p(z_1|\theta)p(x_1|z_1,\theta) \prod_{t=2}^{T} p(z_t|z_{t-1},\theta)p(x_t|z_t,\theta)) \\
&= \log p(z_1|\pi) + \sum_{t=2}^{T} \log p(z_t|z_{t-1},A) + \sum_{t=1}^{T} \log p(x_t|z_t,B)
\end{aligned}$$

在 **M 步**，我们可以对不同参数分别考虑优化。对于转移概率矩阵 A，有

$$\begin{aligned}
HF(A) &= \sum_{\mathbf{Z}} p(\mathbf{x},\mathbf{Z}|\theta_{old}) \sum_{t=2}^{T} \log p(z_t|z_{t-1},A) \\
&= \sum_{t=2}^{T} \sum_{\mathbf{Z}} p(\mathbf{x},\mathbf{Z}|\theta_{old}) \log p(z_t|z_{t-1},A) \\
&= \sum_{t=2}^{T} \sum_{l=1}^{S} \sum_{k=1}^{S} p(Z_t=s_k, Z_{t-1}=s_l, \mathbf{x}|\theta_{old}) \log p(s_k|s_l,A) \\
&= \sum_{t=2}^{T} \sum_{l=1}^{S} \sum_{k=1}^{S} p(Z_t=s_k, Z_{t-1}=s_l, \mathbf{x}|\theta_{old}) \log a_{lk}
\end{aligned}$$

其中，对于元素 a_{ij}，有

$$HF(a_{ij}) = \sum_{t=2}^{T} p(Z_t=s_j, Z_{t-1}=s_i, \mathbf{x}|\theta_{old}) \log a_{ij}$$

为简便起见，记

$$\gamma_{ij} = \sum_{t=2}^{T} p(Z_t=s_j, Z_{t-1}=s_i, \mathbf{x}|\theta_{old}) \tag{11.10}$$

γ_{ij} 的表达式只和上一轮迭代的参数有关，和当前优化变量无关。因此，对于矩阵 A 的第 i 行 $a_{i,:}$，有

$$HF(a_{i,:}) = \sum_{j=1}^{S} \gamma_{ij} \log a_{ij}$$

此外还有约束条件

$$\sum_{j=1}^{S} a_{ij} = 1$$

因此优化问题为

$$\begin{aligned} \max_{a_{i,:}} HF(a_{i,:}) = & \sum_{j=1}^{S} \gamma_{ij} \log a_{ij} \\ \text{s.t.} & \sum_{j=1}^{S} a_{ij} = 1 \end{aligned}$$

构造 Lagrange 函数

$$\mathcal{L}(a_{i,:}) = \sum_{j=1}^{S} \gamma_{ij} \log a_{ij} + \lambda(1 - \sum_{j=1}^{S} a_{ij})$$

求偏导，有

$$\frac{\partial \mathcal{L}}{\partial a_{ij}} = \frac{\gamma_{ij}}{a_{ij}} - \lambda = 0$$

因此

$$a_{ij} = \frac{\gamma_{ij}}{\lambda}$$

由约束条件，有

$$\sum_{j=1}^{S} \frac{\gamma_{ij}}{\lambda} = 1$$

即

$$\lambda = \sum_{j=1}^{S} \gamma_{ij}$$

因此有

$$a_{ij} = \frac{\gamma_{ij}}{\sum_{k=1}^{S} \gamma_{ik}} \tag{11.11}$$

同理，对于 π_i，我们有

$$HF(\pi) = \sum_{k=1}^{S} p(Z_1 = s_k, \mathbf{x}|\theta_{old}) \log \pi_k$$

$$HF(\pi_i) = p(Z_1 = s_i, \mathbf{x}|\theta_{old}) \log \pi_i$$

$$\begin{aligned} \max_{\pi} HF(\pi) = & \sum_{i=1}^{S} p(Z_1 = s_i, \mathbf{x}|\theta_{old}) \log \pi_i \\ \text{s.t.} & \sum_{j=1}^{S} \pi_j = 1 \end{aligned}$$

$$\pi_i = \frac{p(Z_1 = s_i, \mathbf{x}|\theta_{old})}{p(\mathbf{x}|\theta_{old})} \tag{11.12}$$

对于 b_{ij}，我们有

$$HF(B) = \sum_{t=1}^{T}\sum_{k=1}^{S} p(x_t|z_t = s_k, \theta_{old}) \log p(x_t|z_t, B)$$

$$HF(B_{ij}) = \sum_{t=1}^{T} p(x_t = v_j|z_t = s_i, \theta_{old}) I(x_t = v_j) \log p(x_t = v_j|z_t = s_i, B)$$

$$\eta_{ij} = \sum_{t=1}^{T} I(x_t = v_j) p(x_t = v_j|z_t = s_i, \theta_{old}) \tag{11.13}$$

$$\begin{aligned} \max_{b_{i,:}} HF(b_{i,:}) = & \sum_{j=1}^{V} \eta_{ij} \log b_{ij} \\ \text{s.t.} & \sum_{j=1}^{V} b_{ij} = 1 \end{aligned}$$

$$b_{ij} = \frac{\eta_{ij}}{\sum_{k=1}^{V} \eta_{ik}} \tag{11.14}$$

在 **E 步**，我们根据已有的参数 $A_{old}, B_{old}, \pi_{old}$ 计算 M 步中的表达式。回顾解决模型评估任务时定义的前向和后向概率

$$\begin{aligned} \alpha_t(i) =& p(Z_t = s_i, x_1, x_2, ..., x_t),\ t = 1, 2, ..., T \\ \beta_t(i) =& p(x_{t+1}, x_{t+2}, ..., x_T|Z_t = s_i),\ t = 1, 2, ..., T \end{aligned}$$

我们有

$$\begin{aligned} \gamma_{ij} =& \sum_{t=2}^{T} \alpha_{t-1}(i) a_{ij} b_j(x_t) \beta_t(j) \\ \eta_{ij} =& \sum_{t=2}^{T} \alpha_t(i) \beta_t(j) \\ p(Z_1 = s_i, \mathbf{x}|\theta_{old}) =& \pi(i) \beta_1(i) b_1(x_1) \\ p(\mathbf{x}|\theta_{old}) =& \sum_{i=1}^{S} \pi(i) \beta_1(i) b_1(x_1) \end{aligned} \tag{11.15}$$

综合起来，我们就得到了 HMM 参数学习的 E-M 算法。这个算法也叫做 **Baum-Welch 算法**。使用这个算法，每轮迭代中，计算 α 和 β 函数的复杂度各为 $O(TS^2)$。计算中间结果 γ 和 η 的复杂度分别为 $O(TS^2)$ 和 $)(TSV)$。因此一轮更新中总的计算复杂度为 $O(TS(3S + V))$，优于暴力算法的 $O(TS^T)$。

11.3 条件随机场 CRF

条件随机场是一种马尔科夫随机场的变种。它假设一组变量为观测变量 X，一组变量为隐变量 Z。隐变量之间有图表示的节点连接关系。在所有观测变量的条件下，如果隐变量满足局部马尔可夫性

$$p(Z_i|Z_{-i}, X) = p(Z_i|N(Z_i), X), i = 1, 2, ..., n$$

则称这组变量构成条件随机场。和马尔可夫随机场类似，条件随机场也有最大团分解的定理

$$p(\mathbf{Z}|\mathbf{X}) = \frac{1}{Z}\prod_{i=1}^{K}\phi_{C_i}(\mathbf{Z}_{C_i}|\mathbf{X}) \tag{11.16}$$

一般我们取势函数的形式为一组事先定义的特征函数的加和-指数形式，即

$$\phi_C(\mathbf{Z}_C|\mathbf{X}) = \exp\{\sum_{i=1}^{K}\lambda_i f_i(\mathbf{Z}, \mathbf{X})\} \tag{11.17}$$

这里面的 $\lambda \in R^K$ 是可学习的参数。特征函数 f_i 可以是确定的函数，也可以是含有可学习参数的函数。

CRF 广泛用于很多领域。例如，在自然语言处理领域，CRF 常用于序列-序列转换任务。此时的观测节点 $\mathbf{X}$ 对应于输入的自然语言序列（或经过编码的序列），隐序列 $\mathbf{Z}$ 对应于每个时刻的标签（如词性，或另一种语言）。此时特征函数一般是人为定义的有语法知识的函数。在计算机视觉领域，CRF 常用于图像分割。观测节点对应于像素或经过编码的特征图，隐节点对应于分割的类别。此时一般将隐节点的连接建模为二维或全连接，特征函数可以取含有带学习参数的光滑函数。

11.3.1 线性链 CRF

为简单起见，我们以线性链 CRF 为例说明 CRF 模型如何解决推断和学习等任务。

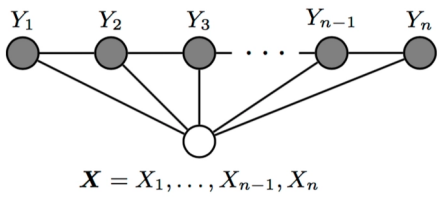

图 11.2: 线性链条件随机场结构

如图11.2所示，线性链条件随机场是一种条件随机场，其中每个隐变量只和上一时刻以及下一时刻有关。根据最大团分解原理，我们可以将其隐变量联合条件概率分布写为

$$p(\mathbf{Z}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})}\phi(Z_1|\mathbf{X})\prod_{t=2}^{T}\phi(Z_t|Z_{t-1}, \mathbf{X})$$

根据上面的定义，我们有 K 个定义在全部观测节点 $\mathbf{X}$ 和相邻隐节点上的特征函数 $f_i(Z_t, Z_{t-1}, \mathbf{X})$，从而有

$$\phi(Z_t|Z_{t-1}, \mathbf{X}) = \sum_{i=1}^{K}\lambda_i f_i(Z_t, Z_{t-1}, \mathbf{X})$$

因此有

$$p(\mathbf{Z}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})}\exp\{\sum_{t=1}^{T}\sum_{i=1}^{K}\lambda_i f_i(Z_t, Z_{t-1}, \mathbf{X})\} \tag{11.18}$$

其中 $Z(\mathbf{X})$ 是配分函数

$$Z(\mathbf{X}) = \int \exp\{\sum_{t=1}^{T}\sum_{i=1}^{K} f_i(Z_t, Z_{t-1}, \mathbf{X})\} d\mathbf{Z} \tag{11.19}$$

11.3.2 配分函数

配分函数的表示比较繁琐。我们可以用矩阵记号简化表达。假设状态 Z_t 的取值范围是有限的 $Z_t \in \mathbf{S} = \{s_1, s_2, ..., s_S\}$，那么可以用矩阵和向量形式表示函数 g, h。记

$$m_{t,ij}(\lambda, \mathbf{x}) = \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(Z_t = s_j, Z_{t-1} = s_i, \mathbf{X} = \mathbf{x})\right\}$$

$$M_t(\lambda, \mathbf{x}) = (m_{t,ij}(\lambda, \mathbf{x}))_{S\times S}$$

考虑定义未归一化的前向概率

$$\alpha_{t,i} = \sum_{Z_1}\sum_{Z_2}...\sum_{Z_t} \exp\{\sum_{i=1}^{K} f_i(Z_t = s_i, Z_{t-1}, \mathbf{X}) \sum_{s=1}^{t-1} f_i(Z_s, Z_{s-1}, \mathbf{X})\}, t = 1, 2, ..., T \tag{11.20}$$

且

$$\alpha_{0,i} = 1 \tag{11.21}$$

这样我们有递推式

$$\alpha_t = M_t(\lambda, \mathbf{x})\alpha_{t-1}, t = 1, 2, ..., T$$

于是有配分函数表达式

$$Z(\mathbf{x}) = \mathbf{1}^T \alpha_T = \mathbf{1}^T M_1(\lambda, \mathbf{x}) M_2(\lambda, \mathbf{x}) ... M_T(\lambda, \mathbf{x})\mathbf{1} \tag{11.22}$$

11.3.3 隐变量推断算法

隐变量推断就是根据一组观测 $\mathbf{x}$ 和已知参数 λ，推测可能性最大的隐状态序列 $\mathbf{z}$ 的过程。由于 CRF 的结构和 HMM 类似，我们也可以使用 Viterbi 算法进行隐变量推断。

需要注意的是，CRF 的 Viterbi 算法里出现的都是未归一化的概率，而不是真实的概率。

11.3.4 学习算法

一般 CRF 的学习任务都是有监督的学习任务，即给定序列 $\mathbf{x}$ 和对应的标注 $\mathbf{z}$，求使得似然最大的参数 λ。如果特征函数有可学习参数，也要一起学习。本小节中为简变起见我们只考虑 λ 的学习。

首先，由于联合分布11.18可以解析表达，配分函数也可以解析计算，因此我们可以使用**梯度上升法**进行最大似然优化。出于篇幅所限，不再赘述。

另一方面，如果定义

$$F_i(\mathbf{X}, \mathbf{Z}) = \sum_{t=1}^{T} f_i(Z_t, Z_{t-1}, \mathbf{X})$$

算法	**条件随机场-Viterbi 算法**
算法简述	对于线性链条件随机场的隐变量推断问题，使用动态规划思想分布求解
已知	样本 $\mathbf{x}=\{x_1,x_2,...,x_T\}$，状态转移矩阵 $A\in R^{S\times S}$， 观测概率矩阵 $B\in R^{S\times V}$，初始分布 $\pi\in R^S$
求	最优隐变量序列 $\arg\max_{\mathbf{Z}} p(\mathbf{Z}\|\mathbf{X},\lambda)$
解类型	闭式解
算法步骤	1. 求初始最可能序列概率 $\delta_1(i)=f_i(s_i,\mathbf{x})$ 2. 迭代求解最可能序列概率 $\delta_t(i)=\max_j f_i(s_i,s_j,\mathbf{x})\delta_{t-1}(j)$ 同时记录前一状态 $c_t(i)=\arg\max_j f_i(s_i,s_j,\mathbf{x})\delta_{t-1}(j)$ 3. $t\leftarrow t+1$, 若 $t<T$ 回 2，否则到 4 4. 计算最优末状态 $i_T=\arg\max_i \delta_T(i)$ 5. 依次求取最优状态 $i_t=c_t(i_{t+1})$，并且 $\hat{z}_t=s_{i_t}$，$t=1,2,...,T$

我们可以将隐变量的联合条件分布11.18写为另一种形式

$$p(\mathbf{Z}|\mathbf{X})=\frac{1}{Z(\mathbf{X})}\exp\{\sum_{i=1}^{K}\lambda_i F_i(\mathbf{X},\mathbf{Z})\}$$

同时

$$Z(\mathbf{X})=\int\exp\{\sum_{i=1}^{K}\lambda_i F_i(\mathbf{X},\mathbf{Z})\}d\mathbf{Z}$$

这和我们在第7.4节推导的最大熵模型的隐变量后验（式7.27）完全相同。只不过此处的隐变量对应于最大熵模型中的标签。考虑对整个数据集进行最大似然

$$\begin{aligned}\max_{\lambda} L(\lambda)&=\sum_{i=1}^{N}\log p(\mathbf{z}^{(i)}|\mathbf{x}^{(i)})\\&=\sum_{i=1}^{N}((\sum_{k=1}^{K}\lambda_k F_k(\mathbf{x}^{(i)},\mathbf{z}^{(i)}))-\log Z(\mathbf{x}^{(i)}))\\&=\sum_{k=1}^{K}\lambda_k\sum_{i=1}^{N}F_k(\mathbf{x}^{(i)},\mathbf{z}^{(i)})-\sum_{i=1}^{N}\log Z(\mathbf{x}^{(i)})\end{aligned}$$

记

$$\tau_k=\sum_{i=1}^{N}F_k(\mathbf{x}^{(i)},\mathbf{z}^{(i)})=\sum_{t=1}^{T}f_k(z_t^{(i)},z_{t-1}^{(i)},\mathbf{x}^{(i)})\tag{11.23}$$

则有

$$L(\lambda)=\sum_{k=1}^{K}\lambda_k\tau_k-\sum_{i=1}^{N}\frac{1}{N}\log\left(\sum_{\mathbf{z}}\exp\left\{\sum_{k=1}^{K}\lambda_k F_k(\mathbf{x}^{(i)},\mathbf{z})\right\}\right)\tag{11.24}$$

该式的形式和最大熵模型的目标函数（式7.29）也完全相同。因此，我们可以直接使用最大熵模型的 IIS 算法进行求解。

算法	**条件随机场-改进迭代尺度法**
算法简述	对于线性链条件随机场的参数学习问题，通过优化似然下界的变化量进行最大似然求解
已知	样本 $X \in R^{T\times N}$，标签 $Z=[\mathbf{z^{(1)}},\mathbf{z^{(2)}},...,\mathbf{z^{(N)}}] \in R^{T\times N}$， 特征函数 $f_i(z_t,z_{t-1},\mathbf{x}), i=1,2,...,K$
求	条件分布 $p(\mathbf{z}\|\mathbf{x},\lambda)$
解类型	迭代解
算法步骤	1. 求出全序列特征函数 $F_i(\mathbf{X},\mathbf{Z})=\sum_{t=1}^{T} f_i(Z_t,Z_{t-1},\mathbf{X})$ 2. 对所有样本和所有标签取值计算全序列补特征函数 $C=\max_k \sum_{i=1}^{K} f_i(\mathbf{x}^{(k)},\mathbf{z}^{(k)})$ $F_{K+1}(\mathbf{x}^{(k)},\mathbf{z}^{(k)})=C-\sum_{i=1}^{K} f_i(\mathbf{x}^{(k)},\mathbf{z}^{(k)})$ 3. 随机初始化参数 $\lambda(0)\in R^{K+1}$ 4. 对每一样本计算模型预测的隐变量分布 $Z_{\lambda(t)}(\mathbf{x})=\int \exp\{\sum_{i=1}^{K}\lambda_i(t)F_i(\mathbf{x},\mathbf{z})\}dy$ $p_{\lambda(t)}(y\|\mathbf{x})=\frac{1}{Z_\lambda(\mathbf{x})}\exp\{\sum_{i=1}^{K}\lambda_i(t)F_i(\mathbf{x},\mathbf{z})\}$ 5. 按下式进行参数迭代 $\delta_i(t)=\frac{1}{C}\log\frac{\sum_{k=1}^{N}F_i(\mathbf{x}^{(k)},\mathbf{z}^{(k)})}{\sum_{j=1}^{N}\sum_z p_{\lambda(t)}(\mathbf{z}\|\mathbf{x}^{(j)})F_i(\mathbf{x}^{(j)},\mathbf{z})}$ $\lambda_i(t+1)=\lambda_i(t)+\delta_i(t)$ 6. $t\leftarrow t+1$, 若参数收敛，停止迭代，否则回到 4.

12 深度生成模型

从第6章开始，我们介绍了许多概率机器学习方法，其中也包含很多**生成式模型**。从概率的观点来看，任何数据 $\mathbf{x}$ 都服从一定的分布 $p(\mathbf{x})$，如图像、音频、视频、文本等。生成式模型的目的就是要获取或建模出这种分布，从而可以得到同一分布之中但不包含在数据集之内的新数据。

对于大部分数据，分布的形式都较为复杂，且数据维度很高。这使得 $p(\mathbf{x})$ 无论是建模还是抽样都不太容易。但在很多情况下，可以认为数据分布由维度较低的**隐变量** $\mathbf{z}$ 控制，即服从隐变量条件分布 $\mathbf{x}\sim p(\mathbf{x}|\mathbf{z})$。这时，先从低维隐变量先验分布 $p(\mathbf{z})$ 中抽样，再根据 $p(\mathbf{x}|\mathbf{z})$ 抽样，那么也相当于从 $p(\mathbf{x})$ 中抽样。

不过，对于真实数据而言，即使是 $p(\mathbf{x}|\mathbf{z})$ 也比较复杂。那么，应当如何对复杂条件分布进行建模呢? 回顾前面章节我们讨论的概率模型，如线性回归 $N(y;\mathbf{w}^T\mathbf{x},\beta^{-1})$ 或 $N(y;\mathbf{w}^T\phi(\mathbf{x}),\beta^{-1})$，我们可以先选取一个已知、简单的分布形式如高斯分布，再使其分布参数被复杂函数 $f(\mathbf{x})$ 建模。因此我们就可以得到形如 $N(\mathbf{x};f(\mathbf{z}),\Sigma)$ 的条件分布。

深度学习时代为我们带来了复杂确定性函数 $f(\mathbf{z})$ 的拟合工具——**深度神经网络**，以及相应的训练学习算法。在本章，我们就介绍一系列使用神经网络建模复杂数据分布的深度生成模型。

12.1 变分自编码器 VAE

变分自编码器是一种使用深度神经网络的生成模型。我们介绍两种不同的推导方式——ELBO 推导和联合分布 KL 散度推导。

算法	**变分自编码器**
算法简述	对于生成问题，假设数据由隐变量生成，后验均服从高斯分布， 使用神经网络拟合分布参数，使用联合分布 KL 散度作为损失
已知	样本 $X \in R^{n\times N}$, 隐变量后验形式 $q(\mathbf{z}\|\mathbf{x},\phi)=N(\mathbf{z};\mu_\phi(\mathbf{x}),\Sigma_\phi(\mathbf{x}))$, 条件分布形式 $p(\mathbf{x}\|\mathbf{z},\theta)=N(\mathbf{x};g_\theta(\mathbf{z}),\sigma_g^2 I)$ ，学习率 α
求	参数 θ,ϕ 使 $q(\mathbf{z}\|\mathbf{x},\phi)\approx p(\mathbf{z}\|\mathbf{x})$, $\hat{p}(\mathbf{x}\|\mathbf{z},\theta)\approx p(\mathbf{x}\|\mathbf{z})$
解类型	迭代解
算法步骤	1. 随机初始化参数 $\theta(0),\phi(0)$ 2. 从均匀分布随机抽取噪声 ϵ 3. 使用编码器神经网络计算噪声分布参数 $\mu_\phi(x),\Sigma_\phi(x)$ 4. 使用重参数化技巧采样 $\mathbf{z}_{\phi,x}=\mu_\phi(\mathbf{x})+\Sigma_\phi(\mathbf{x})\epsilon$ 5. 按下式计算损失函数 $L(\theta,\phi,\mathbf{x})=\frac{1}{2\sigma_g^2}\|\mathbf{x}-g_\theta(\mathbf{z}_{\phi,x})\|^2+\mathrm{tr}(\Sigma_\phi(\mathbf{x}))+\mu_\phi^T\mu_\phi(\mathbf{x})-\sum_{i=1}^K\log\sigma_{i,\phi}^2(\mathbf{x})$ 6. 反向传播更新参数 $\theta(t+1)=\theta(t)-\alpha\frac{\partial L}{\partial\theta}$， $\phi(t+1)=\phi(t)-\alpha\frac{\partial L}{\partial\phi}$ 7. $t\leftarrow t+1$, 若损失函数收敛，停止更新，否则返回 2.

12.1.1 KL 散度推导

本小节的推导参考了苏剑林的博客文章《变分自编码器（二）：从贝叶斯观点出发》。

假设我们要建模隐变量控制的复杂数据分布 $p(\mathbf{x}|\mathbf{z})$。设有参数 θ 控制的神经网络 $g_\theta(z)$，可以拟合一个和 $\mathbf{z}$ 相关的复杂函数。这是一个确定性的函数，而不是一个概率分布。因此我们以它为均值，使用用一个高斯分布作为数据 $\mathbf{x}$ 的条件分布，即

$$p(\mathbf{x}|\mathbf{z})\approx\hat{p}(\mathbf{x}|\mathbf{z},\theta)=N(\mathbf{x};g_\theta(\mathbf{z}),\sigma_g^2 I)$$

其中 σ_g^2 是超参数，事先给定取值。隐变量先验为简单起见，我们设为标准正态分布

$$p(\mathbf{z})=N(\mathbf{0},I)$$

这样我们就可以拟合隐变量和数据的联合分布

$$\hat{p}(\mathbf{x},\mathbf{z}|\theta)=\hat{p}(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{z})$$

按照惯常的概率模型做法，我们应当对联合分布积分得到边缘分布 $\hat{p}(\mathbf{x}|\theta)$，然后最大化对数似然 $\log\hat{p}(\mathbf{x}|\theta)$。

$$\max_\theta\hat{p}(\mathbf{x}|\theta)=\int\hat{p}(\mathbf{x},\mathbf{z}|\theta)d\mathbf{z}$$

然而，由于我们使用了神经网络，并没有办法解析的进行积分。更重要的是，这个模型在训练时无法避免过拟合，因为没有限定 $\mathbf{x}$ 到 $\mathbf{z}$ 的映射关系。在最坏的情况下，模型可能“学会”将隐变量空间中的 N 个点映射到样本集 $\mathbf{x}^{(1)},...,\mathbf{x}^{(N)}$ 中，而其他位置可能映射到无意义的数据，而不是真实的数据分布。

因此，我们有必要限定 $\mathbf{x}$ 到 $\mathbf{z}$ 的映射关系。我们假设条件分布 $p(\mathbf{z}|\mathbf{x})$ 也是一个高斯分布，即

$$p(\mathbf{z}|\mathbf{x})\approx q(\mathbf{z}|\mathbf{x},\phi)=N(\mathbf{z};\mu_\phi(\mathbf{x}),\Sigma_\phi(\mathbf{x}))$$

由于真实的条件分布肯定很复杂，我们还是使用相同的技巧，用神经网络拟合确定性复杂函数，建模分布参数（期望和方差）和条件（$\mathbf{x}$）之间的关系。这里我们假设隐变量条件分布中各维度独立，这样协方差矩阵为对角阵。我们的神经网络

$$\mathbf{f}_\phi(\mathbf{x}) = \begin{bmatrix} \mu_\phi(\mathbf{x}) \\ \Sigma_\phi(\mathbf{x}) \end{bmatrix}$$

实际上输出维度是 $\mathbf{z}$ 的两倍。这样，数据和隐变量间的真实分布也可以进行拟合

$$p(\mathbf{z}, \mathbf{x}) \approx q(\mathbf{z}, \mathbf{x}|\phi) = q(\mathbf{z}|\mathbf{x}, \phi)p(\mathbf{x})$$

上式中虽然 $p(\mathbf{x})$ 未知（而且就是我们要拟合的终极目标），但可以将样本集视作从中采样的样本。因此，如果优化目标可以写成关于 $p(\mathbf{x})$ 的期望的形式，就可以进行优化。

具体来说，我们可以将优化目标选为最小化拟合的联合分布和“真实”的联合分布间的 KL 散度，即

$$\begin{aligned}
&\min_{\theta,\phi} \mathrm{KL}[q(\mathbf{z}, \mathbf{x}|\phi)||\hat{p}(\mathbf{x}, \mathbf{z}|\theta)] \\
=&\int\int q(\mathbf{z}, \mathbf{x}|\phi) \log \frac{q(\mathbf{z}, \mathbf{x}|\phi)}{\hat{p}(\mathbf{x}, \mathbf{z}|\theta)} d\mathbf{z}d\mathbf{x} \\
=&\int p(\mathbf{x}) \int q(\mathbf{z}|\mathbf{x}, \phi) \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)p(\mathbf{x})}{\hat{p}(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z})} d\mathbf{z}d\mathbf{x} \\
=&\left\langle \int q(\mathbf{z}|\mathbf{x}, \phi) \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)p(\mathbf{x})}{\hat{p}(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z})} d\mathbf{z} \right\rangle_{p(\mathbf{x})} \\
=&\left\langle \int -q(\mathbf{z}|\mathbf{x}, \phi) \log \hat{p}(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} \right\rangle + \left\langle \int q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}) d\mathbf{z} \right\rangle \\
&+ \left\langle \int q(\mathbf{z}|\mathbf{x}, \phi) \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} d\mathbf{z} \right\rangle \\
=&\left\langle \int -q(\mathbf{z}|\mathbf{x}, \phi) \log \hat{p}(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} \right\rangle + H[p(\mathbf{x})] + \langle \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z})] \rangle
\end{aligned}$$

其中第二项是真实分布 $p(\mathbf{x})$ 的香农熵，与参数无关。因此我们可以取损失函数为其余两项之和

$$L(\theta, \phi) = \left\langle - \int q(\mathbf{z}|\mathbf{x}, \phi) \log \hat{p}(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} + \mathrm{KL}[q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z})] \right\rangle_{p(\mathbf{x})}$$

由于假设过条件分布的形式，损失函数可以继续化简。第一项为

$$\begin{aligned}
&\left\langle - \int q(\mathbf{z}|\mathbf{x}, \phi) \log \hat{p}(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} \right\rangle_{p(\mathbf{x})} \\
=&\left\langle \left\langle \frac{1}{2\sigma_g^2} ||\mathbf{x} - g_\theta(\mathbf{z})||^2 \right\rangle_{N(\mathbf{z}; \mu_\phi(\mathbf{x}), \Sigma_\phi(\mathbf{x}))} \right\rangle_{p(\mathbf{x})}
\end{aligned}$$

这里内层对隐变量条件分布的期望无法解析求解（因为神经网络无法解析表示），也需要通过抽样的方式估计。然而，抽样是一个随机操作，使得损失函数无法对抽样分布的参数 ϕ 求导。

为解决这个问题，我们引入**重参数化技巧**。假设噪声 $\epsilon \sim N(\epsilon; \mathbf{0}, I)$，则后验分布中抽样的 $\mathbf{z}$ 可以表示为

$$\mathbf{z}_{|\mathbf{x}} = \mu_\phi(\mathbf{x}) + \Sigma_\phi(x)\epsilon$$

相当于将 $\mathbf{z}_{|\mathbf{x}}$ 上的不确定性转移到了 ϵ 上。这样有

$$
\begin{aligned}
&\left\langle -\int q(\mathbf{z}|\mathbf{x},\phi)\log\hat{p}(\mathbf{x}|\mathbf{z},\theta)d\mathbf{z}\right\rangle_{p(\mathbf{x})} \\
=&\left\langle\left\langle \frac{1}{2\sigma_g^2}||\mathbf{x}-g_\theta\left(\mu_\phi(\mathbf{x})+\Sigma_\phi(x)\epsilon\right)||^2\right\rangle_{N(\epsilon;\mathbf{0},I)}\right\rangle_{p(\mathbf{x})}
\end{aligned}
\tag{12.1}
$$

同时，损失的第二项为

$$
\begin{aligned}
&\langle \mathrm{KL}[q(\mathbf{z}|\mathbf{x},\phi)||p(\mathbf{z})]\rangle_{p(\mathbf{x})} \\
=&\left\langle\int q(\mathbf{z}|\mathbf{x},\phi)\log\frac{q(\mathbf{z}|\mathbf{x},\phi)}{p(\mathbf{z})}d\mathbf{z}\right\rangle_{p(\mathbf{x})} \\
=&\left\langle\left\langle -\log|\Sigma_\phi|+\sum_{i=1}^{K}\left(z_i^2-\frac{1}{\sigma_{i,\phi}^2}(z_i-\mu_{i,\phi})^2\right)\right\rangle_{q(\mathbf{z}|\mathbf{x},\phi)}\right\rangle_{p(\mathbf{x})} \\
=&\left\langle\left\langle \sum_{i=1}^{K}\left((1-\frac{1}{\sigma_{i,\phi}^2})z_i^2+\frac{2}{\sigma_{i,\phi}^2}\mu_{i,\phi}z_i-\frac{\mu_{i,\phi}^2}{\sigma_{i,\phi}^2}-\log\sigma_{i,\phi}^2\right)\right\rangle_{q(\mathbf{z}|\mathbf{x},\phi)}\right\rangle_{p(\mathbf{x})} \\
=&\left\langle\sum_{i=1}^{K}\left((1-\frac{1}{\sigma_{i,\phi}^2})(\sigma_{i,\phi}^2+\mu_{i,\phi}^2)+\frac{2}{\sigma_{i,\phi}^2}\mu_{i,\phi}^2-\frac{\mu_{i,\phi}^2}{\sigma_{i,\phi}^2}-\log\sigma_{i,\phi}^2\right)\right\rangle_{p(\mathbf{x})} \\
=&\left\langle\sum_{i=1}^{K}\left(\sigma_{i,\phi}^2+\mu_{i,\phi}^2-1-\log\sigma_{i,\phi}^2\right)\right\rangle_{p(\mathbf{x})}
\end{aligned}
\tag{12.2}
$$

其中 K 为隐变量的维数。因此我们有损失函数

$$
L(\theta,\phi)=\left\langle\left\langle \frac{1}{2\sigma_g^2}||\mathbf{x}-g_\theta\left(\mu_\phi(\mathbf{x})+\Sigma_\phi(x)\epsilon\right)||^2\right\rangle_{N(\epsilon;\mathbf{0},I)}+\sum_{i=1}^{K}\left(\sigma_{i,\phi}^2+\mu_{i,\phi}^2-1-\log\sigma_{i,\phi}^2\right)\right\rangle_{p(\mathbf{x})}
$$

由于会使用随机梯度下降更新参数，对隐变量后验分布的采样也不需要很准确，只采样一次即可，因此

$$
L(\theta,\phi,\mathbf{x})=\frac{1}{2\sigma_g^2}||\mathbf{x}-g_\theta\left(\mu_\phi(\mathbf{x})+\Sigma_\phi(\mathbf{x})\epsilon\right)||^2+\mathrm{tr}(\Sigma_\phi(\mathbf{x}))+\mu_\phi^T\mu_\phi(\mathbf{x})-\sum_{i=1}^{K}\log\sigma_{i,\phi}^2(\mathbf{x}) \tag{12.3}
$$

可以看到，式中前一部分其实就是样本和恢复样本之间的均方误差，后一部分是对隐变量的分布参数的限制，可以看作是隐变量正则化项。模型学习过程就是进行神经网络前向传播——计算损失——梯度反向传播——迭代更新的过程。

在此简单解释一下这个模型的名字。由于神经网络 f_ϕ 和 g_θ 的级联事实上是将高维样本压缩再恢复为高维的过程，可以将其作为一种**自编码器**（AutoEncoder）的变体。而利用 KL 散度度量分布相似性的原理是 KL 散度的非负性，在分布相等时取最小值 0。这个性质的证明来自于**变分法**。

12.1.2 ELBO 推导

上面我们先假设了模型形式，再从联合分布 KL 散度推导出了 VAE。这里我们使用另一种推导方式。

假设现有一组高维复杂分布的数据 $\mathbf{x}$，假设它的生成过程由一定的隐变量 $\mathbf{z}$ 控制，有后验分布 $p(\mathbf{x}|\mathbf{z})$ 和隐变量后验 $p(\mathbf{z}|\mathbf{x})$。

我们假设隐变量后验的近似 $p(\mathbf{z}|\mathbf{x},\theta)$ 是一个高斯分布。参照变分推断 CAVI 的思想，假设其各维度之间相互独立。

$$p(\mathbf{z}|\mathbf{x}) \approx q(\mathbf{z}|\mathbf{x},\theta) = N(\mathbf{z};\mu_\theta(\mathbf{x}),\Sigma_\theta(\mathbf{x}))$$

这个高斯分布的均值和方差应该由 $\mathbf{x}$ 控制，但映射关系很复杂，因此我们假定用一个神经网络来拟合，神经网络参数为 θ，即

$$\mathbf{f}_\theta(\mathbf{x}) = \begin{bmatrix}\mu_\theta(\mathbf{x}) \\ \Sigma_\theta(\mathbf{x})\end{bmatrix}$$

同理，假设样本的后验也服从一个高斯分布，其方差为 $\sigma_g^2 I$，均值为神经网络 $g_\phi(\mathbf{z})$ 的结果

$$p(\mathbf{x}|\mathbf{z}) \approx \hat{p}(\mathbf{x}|\mathbf{z},\phi) = N(\mathbf{x};g_\phi(\mathbf{z}),\sigma_g^2 I)$$

此外，对于隐变量先验 $p(\mathbf{z})$，不妨直接假设其服从标准正态分布

$$p(\mathbf{z}) = N(\mathbf{0}, I)$$

这样我们可以写出对数似然的下界 ELBO（参照式6.6）

$$\log p(\mathbf{x}) \geq \text{ELBO} = \langle \log \hat{p}(\mathbf{x}|\mathbf{z},\theta)\rangle_{q(\mathbf{z}|\mathbf{x},\phi)} - \text{KL}[q(\mathbf{z}|\mathbf{x},\phi)||p(\mathbf{z})]$$

根据式12.2，KL 散度一项为

$$\text{KL}[q(\mathbf{z}|\mathbf{x},\phi)||p(\mathbf{z})] = \sum_{i=1}^{K}\left(\sigma_{i,\phi}^2 + \mu_{i,\phi}^2 - 1 - \log\sigma_{i,\phi}^2\right)$$

而根据式12.1，对数期望一项为

$$\int q(\mathbf{z}|\mathbf{x},\phi)\log\hat{p}(\mathbf{x}|\mathbf{z},\theta)d\mathbf{z} = -\left\langle \frac{1}{2\sigma_g^2}||\mathbf{x} - g_\theta\left(\mu_\phi(\mathbf{x}) + \Sigma_\phi(x)\epsilon\right)||^2\right\rangle_{N(\epsilon;\mathbf{0},I)}$$

这里使用了重参数化技巧，将 $q(\mathbf{z}|\mathbf{x},\phi)$ 采样造成的不确定性转移到噪声 ϵ 上，使得导数可以传递到 μ_ϕ 和 Σ_ϕ。

最大化 ELBO，相当于取损失函数为 ELBO 相反数，即

$$\begin{aligned} L(\theta,\phi) &= -\text{ELBO} \\ &= \left\langle \frac{1}{2\sigma_g^2}||\mathbf{x} - g_\theta\left(\mu_\phi(\mathbf{x}) + \Sigma_\phi(x)\epsilon\right)||^2\right\rangle_{N(\epsilon;\mathbf{0},I)} + \sum_{i=1}^{K}\left(\sigma_{i,\phi}^2 + \mu_{i,\phi}^2 - 1 - \log\sigma_{i,\phi}^2\right)\end{aligned}$$

由于使用梯度下降，因此可以只对噪声采样一次，即

$$L(\theta,\phi) = \frac{1}{2\sigma_g^2}||\mathbf{x} - g_\theta\left(\mu_\phi(\mathbf{x}) + \Sigma_\phi(x)\epsilon\right)||^2 + \sum_{i=1}^{K}\left(\sigma_{i,\phi}^2 + \mu_{i,\phi}^2 - 1 - \log\sigma_{i,\phi}^2\right) \tag{12.4}$$