Subject: Stanford CS229 Machine Learning, Lecture 5, Gaussian discriminant analysis, Naive Bayes

Date: from December 14, 2024 to December 16, 2024

Contents

CS229 Machine Learning, Gaussian discriminant analysis, Naive Bayes, 2022, Lecture 5

YouTube: Stanford CS229 Machine Learning, Gaussian discriminant analysis, Naive Bayes, 2022, Lecture 5

Introduction

Generative learning algorithms – Introduction

生成学习算法/生成模型(Generative learning algorithms / Generative model)是一种建模可观察变量$\boldsymbol{X}$ 和目标变量Y 的联合概率分布$\mathbb{P}(\boldsymbol{X}, Y)$ 的统计模型，在预测时生成模型可用于“生成”新的观察$\boldsymbol{x}$ 的随机实例[a][b].

生成学习算法主要包括两方面：

1. Gaussian Discriminative analysis (GDA) – 高斯判别分析
2. Naive Bayes – 朴素贝叶斯

在前面的Lecture 中所讲的模型都是Discriminative learning algorith，因为我们一直在对模型参数化，然后试图寻找最优的参数，例如在Exponential family 模型中

$$y|\boldsymbol{x};\theta \sim \text{Exponential family}(\eta), \quad \eta = \theta^T\boldsymbol{x}$$

其中θ是参数.

[a]更多介绍可见Wikipedia的Generative model

[b]注意这里的生成模型仅仅是一个统计机器学习算法，而非目前很火的生成式人工智能等等

Generative learning algorithms

Model: 在Generative learning algorithms 中，我们希望建模/参数化(Model / Parameterize) 特征$\boldsymbol{x}$与标签y的联合概率分布$\mathbb{P}(\boldsymbol{x}, y)$，根据条件概率公式，我们有：

$$\mathbb{P}(\boldsymbol{x}, y) = \mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(y) \tag{1}$$

其中y是所有的标签，一般考虑离散情形，因此如果标签有N个类别，那么就需要建模N个式(1).

Learning Time: 在训练时，我们需要学习分布$\mathbb{P}(\boldsymbol{x}|y)$和$\mathbb{P}(y)$，其中$\mathbb{P}(y)$是label 的先验分布(prior)，一般通过假设/观察确定一种分布，再用参数描述，最后通过学习得到.

Testing Time: 在测试时，我们仍是预测给定特征$\boldsymbol{x}$的相应标签y，本质上是计算条件概率$\mathbb{P}(y|\boldsymbol{x})$。根据贝叶斯公式(Bayes Rule)和全概率公式(Law of total probability)，有:

$$\mathbb{P}(y|\boldsymbol{x}) = \frac{\mathbb{P}(\boldsymbol{x}, y)}{\mathbb{P}(\boldsymbol{x})} = \frac{\mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(y)}{\mathbb{P}(\boldsymbol{x})} = \frac{\mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(y)}{\sum_{y'}\mathbb{P}(\boldsymbol{x}|y')\mathbb{P}(y')} \tag{2}$$

根据式(1)，训练阶段已经学习了各个标签的分布$\mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(y)$，因此只需要根据式(2)计算得到每个标签的$\mathbb{P}(y|\boldsymbol{x})$，然后选择概率最大的标签作为预测即可.

Gaussian Discriminative analysis(GDA)

Gaussian Discriminative analysis(GDA) – Introduction

在高斯判别分析中，我们考虑高维情形，即$\boldsymbol{x} \in \mathbb{R}^d$，并且为方便我们规定第一个坐标$x_0 = 1$.

⋆**Assumption**[1]**:** 假设对于各个标签y，$\mathbb{P}(\boldsymbol{x}|y)$满足高维高斯分布，即$\mathbb{P}(\boldsymbol{x}|y) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$.

Question: 为什么不将所有的特征$\boldsymbol{x}$ 统一建模成一个高斯分布，而是对于每一个标签分别建立一个高斯分布呢?

Answer: 在实际问题中，不同类别的样本通常具有不同的分布特征。将所有的特征$\boldsymbol{x}$统一建模为一个高斯分布往往过于简单，无法有效捕捉不同类别之间的差异。

[1] 这个假设是高斯判别分析的重要前提，关于高维高斯分布可见附录A

Gaussian Discriminative analysis(GDA) – 2-Classification Case

GDA – 2-Classification Case – Problem and Model

Problem: 考虑一个二分类问题(如图1)，其中训练集为$\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^n$，标签$y \in \{0, 1\}$.

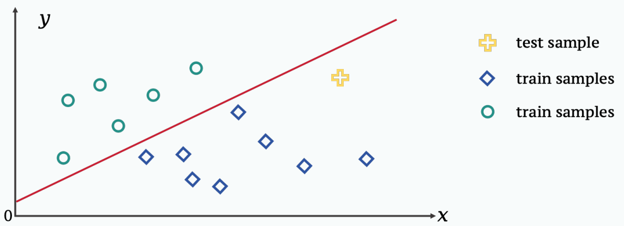


Figure 1: Classification

Model: 首先需要假设两个标签所对应的特征的分布都满足高维高斯分布，即[a]

$$x|(y=0) \sim \mathcal{N}(\boldsymbol{\mu_0}, \Sigma), \quad \boldsymbol{\mu_0} \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d\times d}$$
$$x|(y=1) \sim \mathcal{N}(\boldsymbol{\mu_1}, \Sigma), \quad \boldsymbol{\mu_1} \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d\times d}$$

由于是二分类问题，因此标签 y 的先验概率设为参数为 ϕ 的 Bernoulli 分布，即

$$y \sim \text{Bernoulli}(\phi), \quad \mathbb{P}(y=1) = \phi, \quad \mathbb{P}(y=0) = 1-\phi$$

因此在此模型中参数有四个，为；$\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi$.

[a] 为推导方便设定两个协方差矩阵相同，但当他们不同时也是完全可以推导的，只是复杂一些

GDA – 2-Classification Case – Learn / Fit parameters

Learn / Fit parameters

Before: 为学习模型参数，之前的原则是使得训练集中所有特征-标签对$\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^n$在这些参数下的联合概率最大，即最大化似然(maximum likelihood estimation, MLE):

$$\begin{aligned}
L(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) &= \mathbb{P}((\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \cdots, (\boldsymbol{x}^{(n)}, y^{(n)}); \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) \\
&\overset{i.i.d.}{=} \prod_{i=1}^{n} \mathbb{P}((\boldsymbol{x}^{(i)}, y^{(i)}); \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) \\
&= \prod_{i=1}^{n} \mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}; \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) \cdot \mathbb{P}(y^{(i)}; \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) \\
&= \prod_{i=1}^{n} \mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}; \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma) \cdot \mathbb{P}(y^{(i)}; \phi)
\end{aligned} \tag{3}$$

其中最后一步化简是因为第一项条件概率与ϕ无关，而后一项标签的概率只与ϕ有关.

GDA: 在GDA 中，与以往不同的是我们需要最大化条件概率的似然:

$$\begin{aligned}
L(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) &= \mathbb{P}(y^{(1)}, y^{(2)}, \cdots, y^{(n)}|\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)}; \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) \\
&\overset{i.i.d.}{=} \prod_{i=1}^{n} \mathbb{P}(y^{(i)}|\boldsymbol{x}^{(i)}; \boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi)
\end{aligned} \tag{4}$$

可以看出，在GDA 中我们更关心在观测到$\boldsymbol{x}$后y 的概率，而并不对$\boldsymbol{x}$ 进行单独建模。

GDA – 2-Classification Case – Solutions

Optimize: 与前面Lecture 中一样，在式(3)中最大化似然函数等价于最大化对数似然:

$$\begin{aligned}
&\arg\max L(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) = \arg\max \log\left(L(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi)\right) \triangleq \arg\max l(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) \\
&= \arg\max \sum_{i=1}^{n} \left[\log \mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}) + \log \mathbb{P}(y^{(i)})\right]
\end{aligned} \tag{5}$$

此时只需令$\nabla l(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) = 0$，即[a]

$$\frac{\partial l}{\partial \phi} = 0, \frac{\partial l}{\partial \boldsymbol{\mu_0}} = 0, \frac{\partial l}{\partial \boldsymbol{\mu_1}} = 0, \frac{\partial l}{\partial \Sigma} = 0 \tag{6}$$

Solutions: 首先为将不同标签对应的特征区分开，先定义两个指标集合:

$$U_0 = \{i : y^{(i)} = 0\}, \quad U_1 = \{i : y^{(i)} = 1\}$$

那么根据式(6)最终可以解出(证明详见附录B):

$$\phi = \frac{|U_1|}{n} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(y^{(i)} = 1) \tag{7}$$

$$
\begin{aligned}
\boldsymbol{\mu_0} &= \frac{1}{|U_0|}\sum_{i\in U_0}\boldsymbol{x}^{(i)} = \frac{1}{\sum_{i=1}^{n}\mathbb{I}(y^{(i)}=0)}\left(\sum_{i=1}^{n}\boldsymbol{x}^{(i)}\cdot\mathbb{I}(y^{(i)}=0)\right)\\
\boldsymbol{\mu_1} &= \frac{1}{|U_1|}\sum_{i\in U_1}\boldsymbol{x}^{(i)} = \frac{1}{\sum_{i=1}^{n}\mathbb{I}(y^{(i)}=1)}\left(\sum_{i=1}^{n}\boldsymbol{x}^{(i)}\cdot\mathbb{I}(y^{(i)}=1)\right)
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
\Sigma &= \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}^{(i)}-\boldsymbol{\mu}_{y^{(i)}}\right)\left(\boldsymbol{x}^{(i)}-\boldsymbol{\mu}_{y^{(i)}}\right)^T\\
&= \frac{1}{n}\left[\sum_{i\in U_0}\left(\boldsymbol{x}^{(i)}\right)\left(\boldsymbol{x}^{(i)}\right)^T+\sum_{i\in U_1}\left(\boldsymbol{x}^{(i)}-\boldsymbol{\mu}_1\right)\left(\boldsymbol{x}^{(i)}-\boldsymbol{\mu}_1\right)^T\right]
\end{aligned}
\tag{9}
$$

[a]这四个方程的维数都是与相应参数相对应

GDA – 2-Classification Case – Prediction

Prediction: 给定一个$\boldsymbol{x}$，需要输出$y\in\{0,1\}$，而此时我们的输出为$\arg\max\mathbb{P}(y|\boldsymbol{x})$，实际上只有如下两种可能：

$$
\arg\max\{\mathbb{P}(y=0|\boldsymbol{x};\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi),\mathbb{P}(y=1|\boldsymbol{x};\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi)\}
$$

根据贝叶斯公式可以得到(证明见附录C，可作为练习)

$$
\begin{aligned}
\mathbb{P}(y=1|\boldsymbol{x};\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi) &= \frac{\mathbb{P}(\boldsymbol{x}|y=1;\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma)\cdot\mathbb{P}(y=1;\phi)}{\mathbb{P}(\boldsymbol{x};\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi)}\\
&= \frac{1}{1+\exp\left[-(\theta^T\boldsymbol{x}+\theta_0)\right]},\qquad \theta\in\mathbb{R}^d,\theta_0\in\mathbb{R},\text{均与参数}\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi\text{相关}
\end{aligned}
\tag{10}
$$

Decision Boundary: 不妨记$a=\mathbb{P}(y=0|\boldsymbol{x};\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi),b=\mathbb{P}(y=1|\boldsymbol{x};\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi)$，显然有$a+b=1$，那么

$$
\max\{a,b\}=\begin{cases}a, & \text{if } a\geqslant 0.5>b\\ b, & \text{if } b\geqslant 0.5>a\end{cases}
$$

当$a=b$时，我们称此时的特征集合$\{\boldsymbol{x}:\mathbb{P}(y=0|\boldsymbol{x})=0.5\}$为决策边界(Decision Boundary).

由式(10)可得：

$$
\begin{aligned}
\text{Decision boundary: } &\frac{1}{1+\exp\left[-(\theta^T\boldsymbol{x}+\theta_0)\right]}=0.5\\
\Leftrightarrow \exp&\left[-\left(\theta^T\boldsymbol{x}+\theta_0\right)\right]=1\Leftrightarrow\theta^T\boldsymbol{x}+\theta_0=0
\end{aligned}
\tag{11}
$$

因此如果判定$\boldsymbol{x}$的类别是$y=1$，那么就等价于：

$$
\mathbb{P}(y=1|\boldsymbol{x})>0.5\Leftrightarrow\theta^T\boldsymbol{x}+\theta_0>0 \tag{12}
$$

事实上，当数据不满足高斯性时，也有可能最后得到相同形式的 Decision Boundary (10). 例如$x\in\mathbb{N},x|(y=i)\sim\text{Possion}(\lambda_i),\mathbb{P}(x=k)=e^{-\lambda_i}\frac{\lambda_1^k}{k!},i\in\{0,1\},\mathbb{P}(y=1)=\phi$，此时仍然可以得到式(10).

Summary

Questions and Answers

Q1: 可以看到在二分类 GDA 中只需要求解 Decision Boundary = $\{\boldsymbol{x} : \theta^T \boldsymbol{x} + \theta_0 = 0\}$，那么对于多分类也有 Decision Boundary 吗?

A1: 有，但是会更加复杂，需要仔细判定和设计.

Q2: 在逻辑回归中我们的形式和式(10)是一样的（均为线性判别器），那么这两个模型有什么区别呢?

A2:

	GDA(Generative)	Logistic(Discriminative)
Assumption	$\boldsymbol{x}\|(y=k) \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma), k \in \{0,1\}$ y ~Bernoulli (假设分布)	$\mathbb{P}(y=1\|\boldsymbol{x}) = \frac{1}{1+\exp -\theta^T \boldsymbol{x}}$ (直接假设模型)
Modeling	对$\mathbb{P}(\boldsymbol{x}, y)$建模，由条件概率公式有 $\mathbb{P}(\boldsymbol{x}, y) = \mathbb{P}(\boldsymbol{x}\|y)\mathbb{P}(y)$	仅对$\mathbb{P}(y\|\boldsymbol{x})$建模
Process	模型先学参数 $\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi$，再计算得到$\theta, \theta_0$	模型直接学习 θ

Table 1: Comparison between GDA and Logistic regression

High level perspective

可以看到相较于 Logistic regression，GDA有更多的假设（高斯性），更多的正确的假设会带来更好的模型表现，因为引入了正确的先验知识。然而，引入假设都伴随着假设错误的风险，因此也有风险使模型表现很糟糕.

Good: More assumption + Correct Assumption $\Rightarrow$ Better performance
Risk: You might make wrong assumption! $\Rightarrow$ Worse performance

不同的生成式学习算法(Generative learning algorithm)互相间性能的差异很大程度上是由其假设与问题的贴合性导致的；生成式学习算法与判别式学习算法(Discriminative learning algorith)性能的差异往往是由生成式学习算法做出了好的/不好的假设导致.

在解决问题时，模型获取知识的来源有两个：1. Assumption, 2. Data. 当数据足够多时，有时做先验假设引入的风险反而使做假设引入知识变得不值得，因此在现代的深度学习/大规模机器学习中，数据量已经很大，往往先进的算法已不再有很多的假设。但是在一些特殊领域，例如医疗领域，数据量并不大，因此GDA这些方法仍然奏效.

此外，不同的问题可能需要仔细地根据问题“定制”假设，这需要一些行业经验才能做到。在现代的机器学习/深度学习中，GDA 的使用不如以前那么多，因为现在很多任务甚至都没有标签，例如只有图像或者无标记文本（语言模型）作为$\boldsymbol{x}$.

A Multivariate Gaussian Distribution

Multivariate Gaussian Distribution

高维高斯分布（Multivariate Gaussian Distribution）是高维空间中常见的概率分布。对于一个d-维的随机向量$\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top$，其概率密度函数（PDF）可以表示为:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

其中，

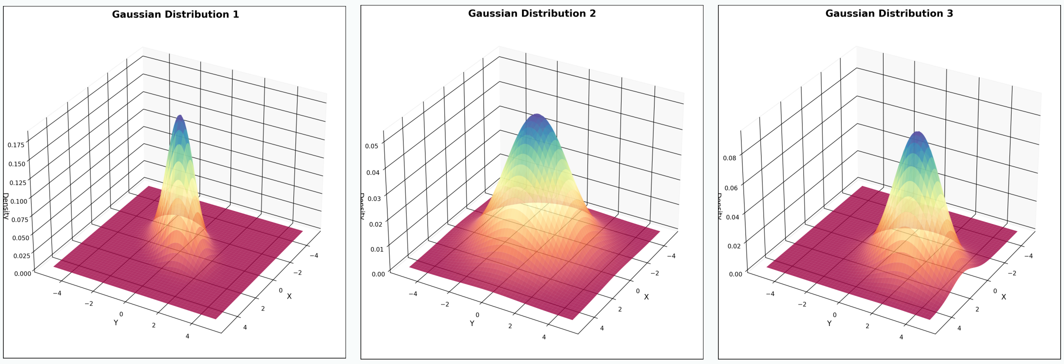
$$\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}] \in \mathbb{R}^d, \quad \boldsymbol{\Sigma} = \mathbb{E}\left[(\boldsymbol{x}-\mathbb{E}[\boldsymbol{x}])(\boldsymbol{x}-\mathbb{E}[\boldsymbol{x}])^T\right] \in \mathbb{R}^{d\times d}$$

$\boldsymbol{\mu}$ 是均值向量，表示高斯分布的中心，$\boldsymbol{\Sigma}$ 是协方差矩阵，描述了各维度之间的线性依赖关系。具体来说，协方差矩阵$\boldsymbol{\Sigma}$ 的元素σ_{ij} 表示第i 维与第j 维的协方差。

协方差矩阵$\boldsymbol{\Sigma}$ 不仅描述了各个维度之间的相关性，还决定了分布的形状。协方差矩阵是对角阵意味着各维度之间是独立的，分布在每一维度上是独立的高斯分布。若协方差矩阵是满秩的，则各维度之间可能存在相关性，分布的形状通常是椭圆形的。(见图A)

最大似然估计：在给定一组样本数据$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ 的情况下，高维高斯分布的最大似然估计（MLE）可以通过样本均值和样本协方差矩阵来得到。具体而言，均值向量和协方差矩阵的估计分别为:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i, \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^\top$$



$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad \mu_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \qquad \mu_3 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1.5 & 0.3 \\ 0.3 & 2 \end{bmatrix}$$

Figure 2: Multivariate Gaussian Distribution

B Proof of the solutions of GDA(2-classification case)

Proof

我们的证明目标是式(7),(8),(9).

Proof. 对数似然函数$l(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi)$ 为:

$$l(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) = \sum_{i=1}^{n} \left[\log \mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}) + \log \mathbb{P}(y^{(i)})\right]$$

其中，$\mathbb{P}(\boldsymbol{x}|y=0) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_0}, \Sigma)$ 和 $\mathbb{P}(\boldsymbol{x}|y=1) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_1}, \Sigma)$，分别是标签为0 和1 时特征的条件概率密度。$\mathbb{P}(y=0) = \phi$ 和$\mathbb{P}(y=1) = 1-\phi$ 分别为标签为0 和1 时的先验概率.

我们需要根据类别$y^{(i)}$ 来决定每个样本的似然:

$$\mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}=0) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})^T \Sigma^{-1} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})\right)$$

$$\mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}=1) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_1})^T \Sigma^{-1} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu_1})\right)$$

因此，总对数似然函数为:

$$l(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) = \sum_{i \in U_0} \left[\log \mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}=0) + \log(1-\phi)\right] + \sum_{j \in U_1} \left[\log \mathbb{P}(\boldsymbol{x}^{(j)}|y^{(j)}=1) + \log(\phi)\right]$$

展开对数似然函数时，我们需要处理每个样本对应的对数概率:

$$\log \mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}=0) = -\frac{1}{2}\log|\Sigma| - \frac{d}{2}\log(2\pi) - \frac{1}{2}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})^T \Sigma^{-1} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})$$

$$\log \mathbb{P}(\boldsymbol{x}^{(i)}|y^{(i)}=1) = -\frac{1}{2}\log|\Sigma| - \frac{d}{2}\log(2\pi) - \frac{1}{2}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_1})^T \Sigma^{-1} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu_1})$$

因此，完整的对数似然函数是:

$$\begin{aligned} l(\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \Sigma, \phi) &= -\frac{n}{2}\log|\Sigma| - \frac{nd}{2}\log(2\pi) + \sum_{i \in U_0}[\log(1-\phi)] + \sum_{j \in U_1}[\log(\phi)] \\ &+ \sum_{i \in U_0}\left[-\frac{1}{2}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})^T \Sigma^{-1} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})\right] + \sum_{j \in U_1}\left[-\frac{1}{2}(\boldsymbol{x}^{(j)} - \boldsymbol{\mu_1})^T \Sigma^{-1} (\boldsymbol{x}^{(j)} - \boldsymbol{\mu_1})\right] \end{aligned}$$

① 对ϕ 求导并令其为零:

$$\frac{\partial l}{\partial \phi} = -\frac{|U_0|}{1-\phi} + \frac{|U_1|}{\phi} = 0 \Rightarrow \phi = \frac{|U_1|}{|U_0| + |U_1|} = \frac{|U_1|}{n} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y^{(i)}=1)$$

② 对$\boldsymbol{\mu_0}$求导并令其为零:

$$\frac{\partial l}{\partial \boldsymbol{\mu_0}} = \frac{\partial}{\partial \boldsymbol{\mu_0}} \sum_{i \in U_0}\left[-\frac{1}{2}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})^T \Sigma^{-1} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})\right] = \sum_{i \in U_0} \Sigma^{-1}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0}) = 0$$

$$\Rightarrow \boldsymbol{\mu_0} = \frac{1}{|U_0|}\sum_{i \in U_0} \boldsymbol{x}^{(i)}$$

③ 同理对$\boldsymbol{\mu_0}$求导并令其为零可得:

$$\boldsymbol{\mu_0} = \frac{1}{|U_1|}\sum_{j\in U_1}\boldsymbol{x}^{(j)}$$

④ 由于$\frac{\partial|\Sigma|}{\partial\Sigma} = |\Sigma|(\Sigma^{-1})^T$，$\frac{\partial\ln|\Sigma|}{\partial\Sigma} = \Sigma^{-T}$，$\frac{\partial\Sigma^{-1}}{\partial\Sigma} = -\Sigma^{-1}\Sigma^{-1}$，因此对$\Sigma$ 求导，得到:

$$\frac{\partial l}{\partial\Sigma} = \frac{1}{2}\left(\sum_{i=1}^{n}\left[(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{y^{(i)}})^T\Sigma^{-1}\Sigma^{-1}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{y^{(i)}})\right] - n\Sigma^{-T}\right)$$

令其等于零，得到:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_{y^{(i)}}})(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_{y^{(i)}}})^T$$

即:

$$\Sigma = \frac{1}{n}\left[\sum_{i\in U_0}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_0})^T + \sum_{i\in U_1}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_1})(\boldsymbol{x}^{(i)} - \boldsymbol{\mu_1})^T\right]$$

□

C Proof of the decision boundary (10)

Proof of (10)

我们的证明目标是:

$$\begin{aligned}\mathbb{P}(y=1|\boldsymbol{x};\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi) &= \frac{\mathbb{P}(\boldsymbol{x}|y=1;\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma)\cdot\mathbb{P}(y=1;\phi)}{\mathbb{P}(\boldsymbol{x};\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi)} \\ &= \frac{1}{1+\exp\left[-\left(\theta^T\boldsymbol{x}+\theta_0\right)\right]}, \quad \theta\in\mathbb{R}^d,\theta_0\in\mathbb{R},\text{均与参数}\boldsymbol{\mu_0},\boldsymbol{\mu_1},\Sigma,\phi\text{相关}\end{aligned} \tag{13}$$

Proof. 根据贝叶斯公式和全概率公式，可以得到

$$\mathbb{P}(y=1|\boldsymbol{x}) = \frac{\mathbb{P}(\boldsymbol{x}|y=1)\mathbb{P}(y=1)}{\mathbb{P}(\boldsymbol{x}|y=1)\mathbb{P}(y=1)+\mathbb{P}(\boldsymbol{x}|y=0)\mathbb{P}(y=0)}$$

同时已经假设

$$\mathbb{P}(y=1)=\phi, \quad \mathbb{P}(y=0)=1-\phi$$

$$\begin{aligned}\mathbb{P}(\boldsymbol{x}|y=1) &= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})\right) \\ \mathbb{P}(\boldsymbol{x}|y=0) &= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_0})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0})\right)\end{aligned}$$

代入可得

$$\begin{aligned}&\mathbb{P}(y=1|\boldsymbol{x}) \\ &= \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})\right)\cdot\phi}{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})\right)\cdot\phi+\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_0})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0})\right)\cdot(1-\phi)} \\ &= \frac{1}{1+\frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_0})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0})\right)\cdot(1-\phi)}{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})\right)\cdot\phi}} \\ &= \frac{1}{1+\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_0})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0})+\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})+\log\frac{1-\phi}{\phi}\right)}\end{aligned}$$

这可以表示为:

$$\mathbb{P}(y=1|\boldsymbol{x}) = \frac{1}{1+\exp\left[-\left(\theta^T\boldsymbol{x}+\theta_0\right)\right]}$$

其中$\theta=\Sigma^{-1}(\boldsymbol{\mu_1}-\boldsymbol{\mu_0})$ 和$\theta_0=\log\frac{\phi}{1-\phi}+\frac{1}{2}\left(\boldsymbol{\mu_0}^T\Sigma^{-1}\boldsymbol{\mu_0}-\boldsymbol{\mu_1}^T\Sigma^{-1}\boldsymbol{\mu_1}\right)$. □

D Codes for Multivariate Gaussian Distribution

Multivariate Gaussian Distribution – Codes

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal
from mpl_toolkits.mplot3d import Axes3D

def plot_3d_gaussian(mu, cov, title="3D Gaussian Distribution", save_path=None):
    """
    Plot the 3D Gaussian distribution

    Parameters:
    mu: Mean, a list of length 2 [mu_x, mu_y]
    cov: Covariance matrix, a 2x2 2D array
    title: Title of the plot
    save_path: Path to save the plot (optional)
    """
    # Generate mesh grid data
    x = np.linspace(-5, 5, 1000)
    y = np.linspace(-5, 5, 1000)
    X, Y = np.meshgrid(x, y)

    # Calculate the probability density of the 2D Gaussian distribution
    pos = np.dstack((X, Y))
    rv = multivariate_normal(mu, cov)
    Z = rv.pdf(pos)

    # Create a 3D surface plot
    fig = plt.figure(figsize=(12, 10), dpi=200)
    ax = fig.add_subplot(111, projection='3d')

    # Plot a smooth surface
    surf = ax.plot_surface(X, Y, Z, cmap="Spectral", edgecolor='none', alpha=0.8)

    # Set title and labels
    ax.set_title(title, fontsize=16, fontweight='bold')
    ax.set_xlabel("X", fontsize=12)
    ax.set_ylabel("Y", fontsize=12)
    ax.set_zlabel("Density", fontsize=12)

    # Set the view angle to avoid a flat view
    ax.view_init(30, 30)

    # Set the grid lines to be black
    ax.grid(True, color='black')  # Show grid lines
    ax.xaxis._axinfo['grid'].update(color='black')  # Grid lines for X axis
    ax.yaxis._axinfo['grid'].update(color='black')  # Grid lines for Y axis
    ax.zaxis._axinfo['grid'].update(color='black')  # Grid lines for Z axis

    # Set the axis tick marks to be black
    ax.tick_params(axis='both', direction='in', length=6, width=1, colors='black')

    # Set the external border lines to be black
    fig.patch.set_edgecolor('black')  # External border lines of the figure
    fig.patch.set_linewidth(2)        # Set the thickness of the border lines
```

```

    # Remove the background color of the panels, making them transparent
    ax.xaxis.pane.fill = True  # Transparent background for X axis
    ax.yaxis.pane.fill = False  # Transparent background for Y axis
    ax.zaxis.pane.fill = False  # Transparent background for Z axis

    # ax.set_facecolor('none')  # Uncomment to make the plotting area background transparent
    # fig.patch.set_alpha(0)  # Uncomment to make the figure background transparent

    # fig.colorbar(surf, shrink=0.5, aspect=5)    # Add a color bar (optional)

    plt.show()  # Display the plot

    plt.draw()  # Force redraw
    plt.savefig(save_path)

# Example calls:
mu1 = [0, 0]
cov1 = [[1, 0], [0, 1]]
plot_3d_gaussian(mu1, cov1, title="Gaussian Distribution 1", save_path="gaussian-1.png")

mu2 = [0, 0]
cov2 = [[1, 0.8], [0.8, 1]]
plot_3d_gaussian(mu2, cov2, title="Gaussian Distribution 2", save_path="gaussian-2.png")

mu3 = [1, 2]
cov3 = [[1.5, 0.3], [0.3, 2]]
plot_3d_gaussian(mu3, cov3, title="Gaussian Distribution 3", save_path="gaussian-3.png")
```

First updated: December 16, 2024
Last updated: December 18, 2024

References