名词解释

1. 机器学习

- 机器学习是一门人工智能的科学，该领域的主要研究对象是人工智能，特别是如何在经验学习中改善具体算法的性能。
- 机器学习是对能通过经验自动改进的计算机算法的研究。
- 机器学习是用数据或以往的经验，以此优化计算机程序的性能标准。

2. 监督学习、无监督学习、半监督学习

- 监督学习(Supervised learning)

通过已有的一部分输入数据与输出数据之间的对应关系，生成一个函数，将输入映射到合适的输出，例如分类。

- 无监督学习(Unsupervised learning)

直接对输入数据集进行建模，例如聚类。

- 半监督学习(Semi-supervised learning)

综合利用有类标的数据和没有类标的数据，来生成合适的分类函数。

3. 线性回归

线性回归（Linear regression）是利用称为线性回归方程的最小二乘函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。只有一个自变量的情况称为简单回归，大于一个自变量情况的叫做多元回归。

4. 逻辑回归

逻辑回归是一种广义的线性回归分析模型，逻辑回归常用于做二分类，

5. 线性回归 VS 逻辑回归

（1）线性回归用来预测，逻辑回归用来分类

（2）线性回归是拟合函数，逻辑回归是预测函数

（3）在参数估计中，线性回归和逻辑回归都可以使用梯度下降算法

6. 代价函数

是指一种将一个事件（在一个样本空间中的一个元素）映射到一个表达与其事件相关的经济成本或机会成本的实数上的一种函数。更通俗地说，损失函数用来衡量参数选择的准确性。

Lecture1：线性回归

1.1 linear hypothesis 线性回归函数

$$
\begin{aligned}
h_\theta(x) &= \theta^T x \\
&= \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n
\end{aligned}
$$

1.2 cost function 代价函数

$$
J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2
$$

1.3 Gradient Descent (GD) Algorithm 梯度下降算法

在GD中，每次迭代都要用到全部训练数据。每次GD的更新算法为

$$
\begin{aligned}
\theta_j &= \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta) \\
&= \theta_j - \alpha\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}
\end{aligned}
$$

1.4 Stochastic Gradient Descent (SGD) 随机梯度下降算法

在SGD中，每次迭代可以只有一个训练数据来更新参数。回到GD的更新算法，假设此时我们从训练数据中随机取一条$(x^{(i)}, y^{(i)})$。此时更新参数的算法变为

$$
\begin{aligned}
\theta_j &= \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta) \\
&= \theta_j - \alpha\frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2 \\
&= \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}
\end{aligned}
$$

1.5 GD VS SGD

当训练集过大时，用GD可能导致内存不够用，那么就可以用SGD了，SGD其实可以算做一种online-learning。另外SGD的收敛速度会比GD的快，它可以防止陷入局部最优解。但是对于代价函数求最小值还是GD做的比较好，不过SGD也够用了。

1.6 matrix derivatives 矩阵求导

- 实值函数y对向量X求导。其中$X \in \mathbb{R}^n$

$$
\frac{\partial y}{\partial X} = \begin{bmatrix}
\frac{\partial y}{\partial x_1} \\
\frac{\partial y}{\partial x_2} \\
\vdots \\
\frac{\partial y}{\partial x_n}
\end{bmatrix}
$$

- 实值函数y对矩阵X求导。其中$X \in \mathbb{R}^{m\times n}$

$$\frac{\partial y}{\partial X}=\begin{bmatrix}\frac{\partial y}{\partial x_{11}} & \cdots & \frac{\partial y}{\partial x_{1n}}\\ \vdots & \ddots & \vdots\\ \frac{\partial y}{\partial x_{m1}} & \cdots & \frac{\partial y}{\partial x_{mn}}\end{bmatrix}$$

1.7 代价函数矩阵化

- 假设

$$X=\begin{bmatrix}(x^{(1)})^T\\ \vdots\\ (x^{(m)})^T\end{bmatrix}\quad Y=\begin{bmatrix}y^{(1)}\\ \vdots\\ y^{(m)}\end{bmatrix}$$

- 因此有

$$X\theta-Y=\begin{bmatrix}(x^{(1)})^T\\ \vdots\\ (x^{(m)})^T\end{bmatrix}-\begin{bmatrix}y^{(1)}\\ \vdots\\ y^{(m)}\end{bmatrix}=\begin{bmatrix}h_\theta(x^{(1)})-y^{(1)}\\ \vdots\\ h_\theta(x^{(m)})-y^{(m)}\end{bmatrix}$$

- 因此代价函数的矩阵化形式为

$$\begin{aligned}J(\theta)&=\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2\\&=\frac{1}{2}(X\theta-Y)^T(X\theta-Y)\end{aligned}$$

1.8 正规方程求解

$$\begin{aligned}\nabla_\theta J(\theta)&=\nabla_\theta\frac{1}{2}(Y-X\theta)^T(Y-X\theta)\\&=\frac{1}{2}\nabla_\theta\left(Y^T-\theta^TX^T\right)(Y-X\theta)\\&=\frac{1}{2}\nabla_\theta\,\mathrm{tr}\left(Y^TY-Y^TX\theta-\theta^TX^TY+\theta^TX^TX\theta\right)\\&=\frac{1}{2}\nabla_\theta\,\mathrm{tr}\left(\theta^TX^TX\theta\right)-X^TY\\&=\frac{1}{2}\left(X^TX\theta+X^TX\theta\right)-X^TY\\&=X^TX\theta-X^TY\end{aligned}$$

其中

$$(1)\nabla_\theta Y^T Y = 0$$
$$(2)\nabla_\theta Y^T X\theta = \nabla_\theta \theta Y^T X = X^T Y$$
$$\text{注：} \quad \operatorname{tr} ABC = \operatorname{tr} CAB = \operatorname{tr} BCA,\ \nabla_A AB = B^T$$
$$(3)\nabla_\theta \theta^T X^T Y = \left(\nabla_{\theta^T} \theta^T X^T Y\right)^T = (Y^T X)^T = X^T Y$$
$$\text{注：} \quad \nabla_{A^T} f(A) = (\nabla_A f(A))^T$$
$$(4)\nabla_\theta \operatorname{tr}(\theta^T X^T X\theta) =$$

令$\nabla_\theta J(\theta) = 0$，得到$\theta = (X^T X)^{-1} X^T Y$。

- $X^T X$不可逆的情况

1. 矩阵的秩 < 矩阵的维度，|A| = 0
2. 向量的个数 > 向量的维数

1.9 正规方程 VS 梯度下降

1. 梯度下降需要选择学习率α，而正规方程不需要
2. 正规方程需要的迭代次数少于梯度下降算法所需的迭代次数
3. 当特征参数相当大的时候，梯度下降法也能够很好的工作
4. 正规方程在计算$X^T X$的逆时，所消耗的时间比较长，当特征参数相当大的时候，计算过程会特别缓慢
5. 梯度下降需要feature scaling，而正规方程不需要

1.10 feature scaling 特征缩放

- **定义**

特征缩放的目标就是数据规范化，使得特征的范围具有可比性。它是数据处理的预处理处理，对后面的使用数据具有关键作用。

- **机器算法为什么要特征缩放**

（1）特征缩放还可以使机器学习算法工作的更好。比如在K近邻算法中，分类器主要是计算两点之间的欧几里得距离，如果一个特征比其它的特征有更大的范围值，那么距离将会被这个特征值所主导。因此每个特征应该被归一化，比如将取值范围处理为0到1之间。

（2）第二个原因则是，特征缩放也可以加快梯度收敛的速度。

- **方法**

（1）Rescaling 调节比例

这种方法是将数据的特征缩放到$[0, 1]$或$[-1, 1]$之间。缩放到什么范围取决于数据的性质。对于这种方法的公式如下：

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

x是最初的特征值，x'是缩放后的值。

（2）Mean normalisation 平均值规范化

$$x' = \frac{x - \operatorname{mean}(x)}{\max(x) - \min(x)}$$

（3）Standardization 标准化

特征标准化使每个特征的值有零均值(zero-mean)和单位方差(unit-variance)。这个方法在机器学习地算法中被广泛地使用。例如：SVM，逻辑回归和神经网络。这个方法的公式如下：

$$x' = \frac{x - \text{mean}(x)}{\text{std}(x)}$$

标准差的定义为

$$\text{std}(x) = \sqrt{\frac{\sum_{i=1}^{n}(x - \text{mean(x)})^2}{n}}$$

（4）Scaling to unit length 缩放到单位长度

$$x' = \frac{x}{||x||}$$

就是除以向量的欧拉长度(the Euclidean length of the vector)，二维范数

Lecture2：逻辑回归

2.1 logistic function(or sigmoid function) 逻辑函数

$$g(z) = 1/(1 + e^{-z})$$

sigmoid function 的一些定理

- 边界： $g(z) \in (0, 1)$
- 对称性 : $1 - g(z) = g(-z)$
- 梯度： $g'(z) = g(z)(1 - g(z))$

2.2 logistic hypothesis 逻辑回归假设

$$h_\theta(x) = g(\theta^T x) = 1/(1 + e^{-\theta^T x})$$

给定输入特征向量x，我们有

$$P(Y = 1|X = x; \theta) = h_\theta(x) = 1/(1 + \exp(-\theta^T x))$$
$$P(Y = 0|X = x; \theta) = 1 - h_\theta(x) = 1/(1 + \exp(\theta^T x))$$

- 逻辑函数的边界

$P(Y = 1|X = x; \theta) = P(Y = 0|X = x; \theta) \quad \Longrightarrow \quad \theta^T x = 0$

2.3 逻辑回归的一般形式

$$p(y|x; \theta) = P(Y = y|X = x; \theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

2.4 极大似然估计

- 定义

最大似然估计，只是一种概率论在统计学的应用，它是参数估计的方法之一。说的是已知某个随机样本满足某种概率分布，但是其中具体的参数不清楚，参数估计就是通过若干次试验，观察其结果，利用结果推出参数的大概值。最大似然估计是建立在这样的思想上：已知某个参数能使这个样本出现的概率最大，我们当然不会再去选择其他小概率的样本，所以干脆就把这个参数作为估计的真实值。

- 离散型

设X为离散型随机变量，$\theta=(\theta_1,\theta_2,\ldots,\theta_k)$为多维参数向量，如果随机变量相互独立，则可得概率函数

$$P(X_1=x_1,\ldots,X_n=x_n)=\prod_{i=1}^{n}p(x_i;\theta_1,\theta_2,\ldots,\theta_k)$$

- 当$\theta=(\theta_1,\theta_2,\ldots,\theta_k)$固定时，上式表示$X_1=x_1,\ldots,X_n=x_n$的概率。
- 当$X_1=x_1,\ldots,X_n=x_n$已知的时候，它又变成$\theta=(\theta_1,\theta_2,\ldots,\theta_k)$的函数，记为

$$L(\theta_1,\theta_2,\ldots,\theta_k)=\prod_{i=1}^{n}p(x_i;\theta_1,\theta_2,\ldots,\theta_k)$$

，称此函数为似然函数。似然函数值的大小意味着该样本值出现的可能性的大小，既然已经得到了样本值$X_1=x_1,\ldots,X_n=x_n$，那么它出现的可能性应该是较大的，即似然函数的值也应该是比较大的，因而最大似然估计就是选择使$L(\theta_1,\theta_2,\ldots,\theta_k)$达到最大值的那个$\theta$作为真实的估计。

2.5 逻辑回归极大似然函数

$$\begin{aligned}L(\theta)&=\prod_i p(y^{(i)}|x^{(i)};\theta)\\&=\prod_i(h_\theta(x^{(i)}))^{y^{(i)}}(1-h_\theta(x^{(i)}))^{1-y^{(i)}}\end{aligned}$$

- 最大对数似然函数

$$l(\theta)=\log L(\theta)=\sum_{i=1}^{m}\left(y^{(i)}\log h_\theta(x^{(i)})+(1-y^{(i)})\log(1-h_\theta(x^{(i)})\right)$$

2.6 逻辑回归参数求解

- 梯度下降算法

$\theta_j\leftarrow\theta_j+\alpha\nabla_{\theta_j}l(\theta)$，对于$\forall j$，其中

$$\frac{\partial}{\partial\theta_j}l(\theta)=\sum_{i=1}^{m}(y^{(i)}-h_\theta(x^{(i)}))x_j^{(i)}$$

推导过程如下：

$$
\begin{aligned}
\nabla_{\theta_j} l(\theta) &= \frac{\partial}{\partial \theta_j} l(\theta) \\
&= \sum_{i=1}^{m} \left(y^{(i)} \frac{1}{g(\theta^T x^{(i)})} - (1 - y^{(i)}) \frac{1}{1 - g(\theta^T x^{(i)})} \right) \frac{\partial}{\partial \theta_j} g(\theta^T x^{(i)}) \\
&= \sum_{i=1}^{m} \left(y^{(i)} \frac{1}{g(\theta^T x^{(i)})} - (1 - y^{(i)}) \frac{1}{1 - g(\theta^T x^{(i)})} \right) g(\theta^T x^{(i)})(1 - g(\theta^T x^{(i)})) \frac{\partial}{\partial \theta_j} \theta^T x^{(i)} \\
&= \sum_{i=1}^{m} \left(y^{(i)} (1 - g(\theta^T x^{(i)})) - (1 - y^{(i)}) g(\theta^T x^{(i)}) \right) x_j^{(i)} \\
&= \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}
\end{aligned}
$$

- 牛顿法求解

$x \leftarrow x - \frac{f(x)}{f'(x)}$

为了最大化$f(x)$, 我们应该使$f'(x) = 0$，此时的牛顿法更新为

$$
x \leftarrow x - \frac{f'(x)}{f''(x)}
$$

即

$$
\theta \leftarrow \theta - H^{-1} \nabla_\theta l(\theta)
$$

其中H是海森矩阵

$$
H_{i,j} = \frac{\partial^2 l(\theta)}{\partial \theta_i \partial \theta_j}
$$

Lecture3：正则化和贝叶斯基础

3.1 Regularization 正则化

模型选择的典型方法是正则化。正则化是结构风险最小化策略的实现，是在经验风险上加一个正则化项(regularizer)或罚项(penalty term)。正则化项一般是模型复杂度的单调递增函数，模型越复杂，正则化值就越大。比如，正则化项可以是模型参数向量的范数。

正则化符合奥卡姆剃刀(Occam's razor)原理。奥卡姆剃刀原理应用于模型选择时变为以下想法：在所有可能选择的模型中，能够很好地解释已知数据并且十分简单才是最好的模型，也就是应该选择的模型。从贝叶斯估计的角度来看，正则化项对应于模型的先验概率。可以假设复杂的模型有较大的先验概率，简单的模型有较小的先验概率。

3.2 解决过拟合现象

- **什么是过拟合**

如果我们有非常多的特征，那么所学的Hypothesis有可能对训练集拟合的非常好，但是对于新数据预测的很差。

- **什么是欠拟合**

函数假设太简单导致无法覆盖足够的原始数据，可能造成数据预测的不准确。

- **减少特征的数量**

（1）人工的选择保留哪些特征

（2）模型选择

- **正则化**

保留所有特征，但降低参数的量/值

正则化的好处是当特征很多时，每一个特征都会对预测y贡献一份合适的力量

3.3 Regularized Linear Regression(线性回归的正则化)

$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)^2+\lambda\sum_{j=1}^{n}\theta_j^2\right]$$

3.4 线性回归正则化求解

- 梯度下降

$\mathbf{Repeat}\{$

$$\begin{aligned}\theta_0&:=\theta_0-\alpha\tfrac{1}{m}\textstyle\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)x_0^{(i)}\\\theta_j&:=\theta_j-\alpha\left[\left(\tfrac{1}{m}\textstyle\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)x_j^{(i)}\right)+\tfrac{\lambda}{m}\theta_j\right]\end{aligned}$$

$\}\mathbf{until}$ convergence condition is satisfied

- 正规方程

$$\theta=\left(X^TX+\lambda\cdot L\right)^{-1}X^T\vec{y}$$

其中

$$L=\begin{bmatrix}0&&&\\&1&&\\&&\ddots&\\&&&1\end{bmatrix}$$

3.5 Regularized Logistic Regression(逻辑回归的正则化)

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right)+\left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right]+\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

3.6 逻辑回归正则化求解

- 梯度下降法

$\mathbf{Repeat}\{$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right) x_0^{(i)}$$
$$\theta_j := \theta_j - \alpha \left(\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right) \text{ for } j = 1, 2, \cdots, n$$

}**until** convergence condition is satisfied

此时，所对应的梯度$\nabla_\theta J(\theta)$和海森矩阵H为

$$\nabla_\theta J(\theta) = \begin{bmatrix} \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)} \\ \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)} + \frac{\lambda}{m} \theta_1 \\ \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)} + \frac{\lambda}{m} \theta_2 \\ \vdots \\ \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_n^{(i)} + \frac{\lambda}{m} \theta_n \end{bmatrix}$$

$$H = \left[\frac{1}{m} \sum_{i=1}^{m} h_\theta(x^{(i)}) (h_\theta(x^{(i)}) - 1) x^{(i)} (x^{(i)})^T \right] + \frac{\lambda}{m} \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

3.7 Maximum Likelihood Estimation (MLE) 极大似然估计

最大似然数，当给定一堆数据$\mathcal{D}$且假定我们已经知道了数据的分布，这个分布的参数θ是固定且未知的。MLE的目标就是找出这样一个固定的参数θ，使得模型产生出现观测数据的概率最大：

$$\theta_{MLE} = \arg\max_\theta p(\mathcal{D}|\theta)$$

若给定m个数据样本且是相互独立的，令$\mathcal{D} = \{d_i\}_{i=1\ldots,m}$，则上述表达式可以替换为

$$\theta_{MLE} = \arg\max_\theta p(\mathcal{D}|\theta) = \arg\max_\theta \prod_{i=1}^{m} p(d_i|\theta)$$

MLE通常最大化对数似然而不是似然

$$\theta_{MLE} = \arg\max_\theta \log p(\mathcal{D}|\theta) = \arg\max_\theta \sum_{i=1}^{m} p(d_i|\theta)$$

- **example**

我们抛硬币，正面朝上的次数满足二项分布，正面朝上的概率是μ。现在我们抛10枚硬币，正面朝上的次数是2，明显$\mu = 0.2$，但现在我们用MLE去求解这个参数

$$p(\mathcal{D}|\mu) = \prod_{i=1}^{n} \mu^{x_i} (1-\mu)^{1-x_i}$$

其中$x_i = 1$表示正面朝上，$x_i = 0$表示反面朝上，n是抛硬币的次数。对数的形式为

$$
\begin{aligned}
\log p(\mathcal{D}|\mu) &= \log\left(\prod_{i=1}^{n}\mu^{x_i}(1-\mu)^{1-x_i}\right) \\
&= \sum_{i=1}^{n}\log\left(\mu^{x_i}(1-\mu)^{1-x_i}\right) \\
&= \sum_{i=1}^{n}\left[\log\mu^{xi} + \log(1-\mu)^{1-xi}\right] \\
&= \sum_{i=1}^{n}\left[x_i\log\mu + (1-x_i)\log(1-\mu)\right]
\end{aligned}
$$

对参数μ求导得

$$
\begin{aligned}
\frac{\partial\log p(\mathcal{D}|\mu)}{\partial\mu} &= \sum_{i=1}^{n}\frac{\partial}{\partial\mu}\left[x_i\log\mu + (1-x_i)\log(1-\mu)\right] \\
&= \sum_{i=1}^{n}\left[x_i\frac{\partial}{\partial\mu}\log\mu + (1-x_i)\frac{\partial}{\partial\mu}\log(1-\mu)\right] \\
&= \frac{1}{\mu}\sum_{i=1}^{n}x_i - \frac{1}{1-\mu}\sum_{i=1}^{n}(1-x_i)
\end{aligned}
$$

我们令导数为0，得到

$$
\frac{2}{\mu} - \frac{8}{1-\mu} = 0
$$

求解方程得$\mu = 0.2$

3.8 Maximum-a-Posteriori Estimation (MAP) 最大化后验估计

最大后验估计和最大似然数类似。当给定一堆数据$\mathcal{D}$且假定我们已经知道了数据的分布，这个分布的参数θ是固定且未知的。唯一和极大似然数不同的是这个参数θ服从某一个分布 ，即参数取到每个值的可能性不是相等的，而是服从一个分布。MAP是根据贝叶斯定理先验转后验推导出来的：

$$
\begin{aligned}
\theta_{MAP} &= \arg\max_{\theta} p(\theta|\mathcal{D}) \\
&= \arg\max_{\theta}\frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \\
&= \arg\max_{\theta} p(\theta)p(\mathcal{D}|\theta) \\
&= \arg\max_{\theta}(\log p(\theta) + \log p(\mathcal{D}|\theta)) \\
&= \arg\max_{\theta}\left(\log p(\theta) + \sum_{i=1}^{m}\log p\left(d_i|\theta\right)\right)
\end{aligned}
$$

- $p(\theta)$: θ的先验概率(没有知道任何数据)
- $p(\mathcal{D})$: 数据的概率(独立于θ)

- **example before**

3.9 线性回归(逻辑回归)中的MLE VS MAP

- MLE参数估计方法会导致非正则解
- MAP参数估计方法会导致正则解
- 先验分布充当MAP估计中的正则化项，也就是不同的先验分布会导致不同的正则解

3.10 逻辑回归中MLE Solution

- 考虑如下方程($y \in \{-1, 1\}$)

$$p(y|x;\theta) = g(y\theta^T x) = \frac{1}{1+\exp(-y\theta^T x)}$$

- Log-likelihood

$$\begin{aligned}\ell(\theta) &= \log \prod_{i=1}^{m} p\left(y^{(i)}|x^{(i)}, \theta\right) \\ &= \sum_{i=1}^{m} \log p\left(y^{(i)}|x^{(i)}, \theta\right) \\ &= \sum_{i=1}^{m} \log \frac{1}{1+\exp\left(-y^{(i)}\theta^T x^{(i)}\right)} \\ &= -\sum_{i=1}^{m} \log\left[1+\exp\left(-y^{(i)}\theta^T x^{(i)}\right)\right]\end{aligned}$$

- MLE solution

$$\theta_{MLE} = \arg\min_{\theta} \sum_{i=1}^{m} \log\left[1+\exp\left(-y^{(i)}\theta^T x^{(i)}\right)\right]$$

没有闭式解存在，但我们可以使用梯度下降算法。

3.11 逻辑回归中MAP Solution

Lecture4：高斯判别式、朴素贝叶斯

4.1 概率回顾

4.1.1 Conditional Probability 条件概率

- 条件概率的定义

$$P(A|B)=\frac{p(A,B)}{p(B)} \quad ,P(A,B)=P(A|B)P(B)$$

- 链式法则

$$P(A_1,A_2,\ldots,A_n)=\prod_{k=1}^{n}P(A_k|A_1,A_2,\ldots,A_{k-1})$$

- 例子

$$P(A_4,A_3,A_2,A_1)=P(A_4|A_3,A_2,A_1)P(A_3|A_2,A_1)P(A_2|A_1)P(A_1)$$

4.1.2 Joint Probability Distribution 联合概率分布

- 假设X和Y都是离散型随机变量
- 随机变量(X,Y)的分布律为

$$p_{X,Y}(x,y)=P(X=x,Y=y)$$

- 联合概率边缘分布

$$p_X(x)=\sum_y P(X=x,Y=y)=\sum_y P(X=x|Y=y)P(Y=y)$$
$$p_Y(y)=\sum_x P(X=x,Y=y)=\sum_x P(Y=y|X=x)P(X=x)$$

4.1.3 Conditional Probability Distribution 条件概率分布

$$p_{Y|X}(y|x)=\frac{p(x,y)}{p_X(x)}, \quad \forall y$$

4.2 Bayes' Theorem 贝叶斯理论

贝叶斯定理描述了事件的概率，基于与事件相关的条件的先验知识

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

- 另一种形式

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

4.3 Gaussian Distribution 高斯分布(mark)

4.3.1 一元高斯分布

- 高斯分布(正态分布)

$$p(x;\mu,\sigma)=\frac{1}{\left(2\pi\sigma^2\right)^{1/2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

4.3.2 Multivariate Gaussian Distribution 多元高斯分布

- 在n维的多元正态分布$\mathcal{N}(\boldsymbol{\mu},\Sigma)$

$$p(\boldsymbol{x};\boldsymbol{\mu},\Sigma)=\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

  - 均值向量$\mu\in\mathbb{R}^n$
  - 协方差矩阵 $\Sigma\in\mathbb{R}^{n\times n}$
  - 马哈拉诺比斯距离：$r^2=(x-\mu)^T\Sigma^{-1}(x-\mu)$

- Σ是半正定的

$$\Sigma=E\{(\boldsymbol{x}-\boldsymbol{\mu})(x-\mu)\}=\Phi\Lambda\Phi^T$$

- Φ是一个正交矩阵，其列是Σ的特征向量
- Λ是对角矩阵，对角元素是特征值

4.3.3 Gaussian Distribution Analysis 高斯判别分析

- $Y\sim Bernoulli(\psi)$
-
  - $P(Y=1)=\psi$
  - $P(Y=0)=1-\psi$
  - Probability mass function

$$p(y)=\psi^y(1-\psi)^{1-y},\quad \forall y=0,1$$

首先，高斯判别分析的作用也是用于分类。对于两类样本，其服从伯努利分布，而对每个类中的样本，假定都服从高斯分布，则有:

$$\begin{aligned}y&\sim\text{Bernouli}(\phi)\\x|y=0&\sim\mathcal{N}(\mu_0,\Sigma)\\x|y=1&\sim\mathcal{N}(\mu_1,\Sigma)\end{aligned}$$

因此，有

$$p(x|y)=\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\right)$$

这样，根据训练样本，估计出先验概率以及高斯分布的均值和协方差矩阵（注意这里两类内部高斯分布的协方差矩阵相同），即可通过如下贝叶斯公式求出一个新样本分别属于两类的概率，进而可实现对该样本的分类。

$$p(y|x)=\frac{p(x|y)p(y)}{p(x)}$$

$$y=\arg\max_y p(y|x)=\arg\max_y \frac{p(x|y)p(y)}{p(x)}=\arg\max_y p(x|y)p(y)$$

- 极大似然对数函数

$$\begin{aligned}
&\ell\left(\psi,\mu_0,\mu_1,\Sigma\right)\\
=&\log\prod_{i=1}^{m}p\left(x^{(i)},y^{(i)};\psi,\mu_0,\mu_1,\Sigma\right)\\
=&\log\prod_{i=1}^{m}p\left(x^{(i)}|y^{(i)};\mu_0,\mu_1,\Sigma\right)p\left(y^{(i)};\psi\right)\\
=&\sum_{i=1}^{m}\log p\left(x^{(i)}|y^{(i)};\mu_0,\mu_1,\Sigma\right)+\sum_{i=1}^{m}\log p\left(y^{(i)};\psi\right)\\
=&\sum_{i=1}^{m}\log\left(p\left(x^{(i)}|y^{(i)}=0\right)^{1-y^{(i)}}*p\left(x^{(i)}|y^{(i)}=1\right)^{y^{(i)}}\right)+\sum_{i=1}^{m}\log p\left(y^{(i)};\psi\right)\\
=&\sum_{i=1}^{m}\left(1-y^{(i)}\right)\log p\left(x^{(i)}|y^{(i)}=0\right)+\sum_{i=1}^{m}y^{(i)}\log p\left(x^{(i)}|y^{(i)}=1\right)+\sum_{i=1}^{m}\log p\left(y^{(i)};\psi\right)
\end{aligned}$$

4.3.4 高斯判别分析求解

由之前所求的极大似然函数，发现函数的第一部分只和μ_0,Σ有关，第二部分只和μ_1,Σ有关，最后一部分只和ψ有关。

- 对ψ求偏导

$$\begin{aligned}
\frac{\partial l\left(\psi,\mu_0,\mu_1,\Sigma\right)}{\partial\psi}&=\frac{\sum_{i=1}^{m}\log p\left(y^{(i)}\right)}{\partial\psi}\\
&=\frac{\partial\sum_{i=1}^{m}\log\psi^{y^{(i)}}(1-\psi)^{1-y^{(t)}}}{\partial\psi}\\
&=\frac{\partial\sum_{i=1}^{m}\left(y^{(i)}\log\psi+\left(1-y^{(i)}\right)\log(1-\psi)\right)}{\partial\psi}\\
&=\sum_{i=1}^{m}\left(y^{(i)}\frac{1}{\psi}-\left(1-y^{(i)}\right)\frac{1}{1-\psi}\right)\\
&=\sum_{i=1}^{m}\left(I\left(y^{(i)}=1\right)\frac{1}{\psi}-I\left(y^{(i)}=0\right)\frac{1}{1-\psi}\right)
\end{aligned}$$

其中I为指示函数。令其为0，可求解出

$$\psi=\frac{\sum_{i=1}^{m}I\left(y^{(i)}=1\right)}{\sum_{i=1}^{m}\left(I\left(y^{(i)}=0\right)+I\left(y^{(i)}=1\right)\right)}=\frac{\sum_{i=1}^{m}I\left(y^{(i)}=1\right)}{m}$$

- 对μ_0和μ_1求偏导

$$\begin{aligned}\frac{\partial l\left(\psi, \mu_0, \mu_1, \Sigma\right)}{\partial \mu_0} &= \frac{\partial\left(1-y^{(i)}\right) \log p\left(x^{(i)} | y^{(i)}=0\right)}{\partial \mu_0} \\ &= \frac{\sum_{i=1}^m\left(1-y^{(i)}\right)\left(\log \frac{1}{(2 \pi)^{n / 2}|\Sigma|^{1 / 2}}-\frac{1}{2}\left(x^{(i)}-\mu_0\right)^T \Sigma^{-1}\left(x^{(i)}-\mu_0\right)\right)}{\partial \mu_0} \\ &= \sum_{i=1}^m\left(1-y^{(i)}\right)\left(\Sigma^{-1}\left(x^{(i)}-\mu_0\right)\right) \\ &= \sum_{i=1}^m I\left(y^{(i)}=0\right) \Sigma^{-1}\left(x^{(i)}-\mu_0\right)\end{aligned}$$

令其为0，求得

$$\mu_0=\frac{\sum_{i=1}^m I\left(y^{(i)}=0\right) x^{(i)}}{\sum_{i=1}^m I\left(y^{(i)}=0\right)}$$

同理，根据对称性，我们可以得到

$$\mu_1=\frac{\sum_{i=1}^m I\left(y^{(i)}=1\right) x^{(i)}}{\sum_{i=1}^m I\left(y^{(i)}=1\right)}$$

- 对Σ求偏导

由于似然函数只有前面两部分和Σ有关，将前两部分改写如下

$$\begin{aligned}&\sum_{i=1}^m(1-y^{(i)}) \log p(x^{(i)} | y^{(i)}=0)+\sum_{i=1}^m y^{(i)} \log p(x^{(i)} | y^{(i)}=1) \\ &=\sum_{i=1}^m(1-y^{(i)})\left(\log \frac{1}{(2 \pi)^{n / 2}|\Sigma|^{1 / 2}}-\frac{1}{2}(x^{(i)}-\mu_0)^T \Sigma^{-1}(x^{(i)}-\mu_0)\right) \\ &\quad+\sum_{i=1}^m y^{(i)}\left(\log \frac{1}{(2 \pi)^{n / 2}|\Sigma|^{1 / 2}}-\frac{1}{2}(x^{(i)}-\mu_1)^T \Sigma^{-1}(x^{(i)}-\mu_1)\right) \\ &=\sum_{i=1}^m\left(\log \frac{1}{(2 \pi)^{n / 2}|\Sigma|^{1 / 2}}-\frac{1}{2}(x^{(i)}-\mu_{y^{(i)}})^T \Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}})\right) \\ &=\sum_{i=1}^m\left(-\frac{n}{2} \log (2 \pi)-\frac{1}{2} \log (|\Sigma|)\right)-\frac{1}{2} \sum_{i=1}^m(x^{(i)}-\mu_{y^{(i)}})^T \Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}})\end{aligned}$$

因此，有

$$\begin{aligned}\frac{\partial \ell\left(\psi, \mu_0, \mu_1, \Sigma\right)}{\partial \Sigma} &= -\frac{1}{2}\sum_{i=1}^{m}\left(\frac{1}{|\Sigma|}|\Sigma|\Sigma^{-1}\right) - \frac{1}{2}\sum_{i=1}^{m}\left(x^{(i)}-\mu_{y^{(i)}}\right)\left(x^{(i)}-\mu_{y^{(t)}}\right)^{T}\frac{\partial \Sigma^{-1}}{\partial \Sigma} \\ &= -\frac{m}{2} - \frac{1}{2}\sum_{i=1}^{m}\left(x^{(i)}-\mu_{y^{(t)}}\right)\left(x^{(i)}-\mu_{y^{(t)}}\right)^{T}\left(-\Sigma^{-2}\right)\end{aligned}$$

这里推导用到了

$$\begin{aligned}\frac{\partial|\Sigma|}{\partial \Sigma} &= |\Sigma|\Sigma^{-1} \\ \frac{\partial \Sigma^{-1}}{\partial \Sigma} &= -\Sigma^{-2}\end{aligned}$$

令其为0，从而求得

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}-u_{y^{(t)}}\right)\left(x^{(i)}-u_{y^{(t)}}\right)^{T}$$

- 最终结果

$$\begin{aligned}\psi &= \frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\left\{y^{(i)}=1\right\} \\ \mu_0 &= \sum_{i=1}^{m}\mathbf{1}\left\{y^{(i)}=0\right\}x^{(i)} / \sum_{i=1}^{m}\mathbf{1}\left\{y^{(i)}=0\right\} \\ \mu_1 &= \sum_{i=1}^{m}\mathbf{1}\left\{y^{(i)}=1\right\}x^{(i)} / \sum_{i=1}^{m}\mathbf{1}\left\{y^{(i)}=1\right\} \\ \Sigma &= \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}-\mu_{y^{(i)}}\right)\left(x^{(i)}-\mu_{y^{(i)}}\right)^{T}\end{aligned}$$

4.3.5 高斯判别分析例子

给定一个测试样本x，我们可以计算

$$\begin{aligned}p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x)} \\ &= \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1)+p(x|y=0)p(y=0)} \\ &= \frac{1}{1+\frac{p(x|y=0)p(y=0)}{p(x|y=1)p(y=1)}}\end{aligned}$$

根据前面的公式

$$
\begin{aligned}
&\frac{p(x|y=0)p(y=0)}{p(x|y=1)p(y=1)} \\
=&\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)+\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\cdot\frac{1-\psi}{\psi} \\
=&\exp\left((\mu_0-\mu_1)^T\Sigma^{-1}x+\frac{1}{2}\left(\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0\right)\right)\cdot\exp\left(\log\left(\frac{1-\psi}{\psi}\right)\right) \\
=&\exp\left((\mu_0-\mu_1)^T\Sigma^{-1}x+\frac{1}{2}\left(\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0\right)+\log\left(\frac{1-\psi}{\psi}\right)\right)
\end{aligned}
$$

4.3.6 GDA分界线

由前面的推导得到

$$
(1-\psi)\exp\big((x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\big)=\psi\exp\big((x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\big)
$$

取对数展开后化简，可得

$$
2x^T\Sigma^{-1}(\mu_1-\mu_0)=\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0+\log\psi-\log(1-\psi)
$$

若

$$
\begin{aligned}
&A=2\Sigma^{-1}(\mu_1-\mu_0)=(a_1,a_2,\ldots,a_n) \\
&b=\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0+\log\psi-\log(1-\psi)
\end{aligned}
$$

因此

$$
a_1x_1+a_2x_2+\cdots+a_nx_n=b
$$

4.3.7 GDA and Logistic Regression 高斯判别分析和逻辑回归

GDA是逻辑回归的一个特例

- **两者的比较：**

  a. 逻辑回归是基于弱假设推导的，其效果更稳定，适用范围更广

  b. 数据服从高斯分布时，GDA的效果更好

  c. 当训练样本很大时，根据中心极限定理，数据将无限逼近于高斯分布，则此时GDA的表现更好些

- **为何假设两类内部高斯分布协方差矩阵相同**

从直观上讲，假设两个类的高斯分布协方差矩阵不同，会更加合理（在混合高斯模型中就是如此假设的），而且可推导出类似上面简洁的结果。

假定两个类有相同协方差矩阵，分析具有以下几点影响：

A．当样本不充分时，使用不同协方差矩阵会导致算法稳定性不够；过少的样本甚至导致协方差矩阵不可逆，那么GDA算法就没法进行

B．使用不同协方差矩阵，最终GDA的分界面不是线性的，同样也推导不出GDA的逻辑回归形式

- **使用GDA时对训练样本有何要求?**

首先，正负样本数的比例需要符合其先验概率。若是预先明确知道两类的先验概率，那么可使用此概率来代替GDA计算的先验概率；若是完全不知道，则可以公平地认为先验概率为50%。

其次，样本数必须不小于样本特征维数，否则会导致协方差矩阵不可逆，按照前面分析应该是多多益善。

4.4 Naive Bayes 朴素贝叶斯

4.4.1 定义

- 训练数据$(x^{(i)}, y^{(i)})_{i=1,\dots,m}$
- - $x^{(i)}$是一个n维向量，$x_j^{(i)} \in \{0,1\}, \quad j=1,\dots,n$
- - $y^{(i)} \in \{1,\dots,k\}$

- 对于$\forall j \neq j'$，朴素贝叶斯假设x_j和$x_{j'}$条件独立

$$
\begin{aligned}
&P\left(X_1=x_1, X_2=x_2, \cdots, X_n=x_n | Y=y\right) \\
=&\prod_{j=1}^{n} P\left(X_j=x_j | X_1=x_1, X_2=x_2, \cdots, X_{j-1}=x_{j-1}, Y=y\right) \\
=&\prod_{j=1}^{n} P\left(X_j=x_j | Y=y\right)
\end{aligned}
$$

- 朴素贝叶斯模型

$$
\begin{aligned}
&P(Y=y|X_1=x_1,\dots,X_n=x_n) \\
=&\frac{P(X_1=x_1,\dots,X_n=x_n|Y=y)P(Y=y)}{P(X_1=x_1,\dots,X_n=x_n)} \\
=&\frac{P(Y=y)\prod_{j=1}^{m}P(X_j=x_j|Y=y)}{P(X_1=x_1,\dots,X_n=x_n)}
\end{aligned}
$$

因此

$$
P(Y=y|X_1=x_1,\dots,X_n=x_n) \propto P(Y=y)\prod_{j=1}^{m}P(X_j=x_j|Y=y)
$$

即

$$
P(y|x_1,\dots,x_n) \propto P(y)\prod_{j=1}^{m}P_j(x_j|y)
$$

- **example**

给定一个新的测试样例，$x=[x_1,x_2,\dots,x_n]$，朴素贝叶斯的输出结果为

$$\arg \max_{y \in \{1,\ldots,k\}} \left(P(y) \prod_{j=1}^{m} P_j(x_j|y) \right)$$

4.4.2 朴素贝叶斯的极大似然函数

$$\begin{aligned}
\ell(\Omega) &= \log \prod_{i=1}^{m} p\left(x^{(i)}, y^{(i)}\right) \\
&= \sum_{i=1}^{m} \log p\left(x^{(i)}, y^{(i)}\right) \\
&= \sum_{i=1}^{m} \log \left(p\left(y^{(i)}\right) \prod_{j=1}^{n} p_j\left(x_j^{(i)}|y^{(i)}\right) \right) \\
&= \sum_{i=1}^{m} \log p\left(y^{(i)}\right) + \sum_{i=1}^{m} \sum_{j=1}^{n} \log p_j\left(x_j^{(i)}|y^{(i)}\right)
\end{aligned}$$

- 极大似然朴素贝叶斯最终模型

$$\max \sum_{i=1}^{m} \log p\left(y^{(i)}\right) + \sum_{i=1}^{m} \sum_{j=1}^{n} \log p_j\left(x_j^{(i)}|y^{(i)}\right)$$

$$s.t \quad \begin{array}{l}
\sum_{y=1}^{k} p(y) = 1 \\
\sum_{x \in \{0,1\}} p_j(x|y) = 1, \forall y, j \\
p(y) \geq 0, \forall y \\
p_j(x|y) \geq 0, \forall j, x, y
\end{array}$$

4.4.3 朴素贝叶斯极大似然函数求解

采用拉格朗日乘子的方法，对应的拉格朗日函数为

$$\begin{aligned}
L(\Omega, \alpha, \beta) = & \sum_{i=1}^{m} \log p\left(y^{(i)}\right) + \sum_{i=1}^{m} \sum_{j=1}^{n} \log p_j\left(x_j^{(i)}|y^{(i)}\right) \\
& - \alpha \left(\sum_{y=1}^{k} p(y) - 1 \right) \\
& - \sum_{y=1}^{k} \sum_{j=1}^{n} \beta_j(y) \left(\sum_{x \in \{0,1\}} p_j(x|y) - 1 \right)
\end{aligned}$$

其中的α和$\beta = \{\beta_j(y)\}_{j \in [n], y \in [k]}$是拉格朗日乘子

- 对$p(y)$求偏导

$$\frac{\partial}{\partial p(y)} L(\Omega, \alpha, \beta) = \sum_{i: y^{(i)} = y} \frac{\partial}{\partial p(y)} \log p(y) - \alpha = \frac{\text{count}(y)}{p(y)} - \alpha = 0$$

其中

$$\text{count}(y)=\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right),\forall y\in[k]$$

求解得

$$p(y)=\frac{\text{count}(y)}{\alpha}$$

根据约束条件$\sum_{x\in\{0,1\}}p_j(x|y)=1$，有

$$\sum_{y=1}^{k}p(y)=\sum_{y=1}^{k}\frac{\text{count}(y)}{\alpha}=\frac{m}{\alpha}=1$$

得到$\alpha=m$，因此

$$p(y)=\frac{\text{count}(y)}{\alpha}=\frac{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)}{m}$$

- 对$p_j(x|y)$求偏导

$$\frac{\partial}{\partial p_j(x|y)}L(\Omega,\alpha,\beta)=\frac{\text{count}_j(x|y)}{p_j(x|y)}-\beta_j(y)=0$$

其中

$$\text{count}_j(x|y)=\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\wedge x_j^{(i)}=x\right),\forall y\in[k],\forall x\in\{0,1\}$$

得到

$$p_j(x|y)=\frac{\text{count}_j(x|y)}{\beta_j(y)}$$

根据约束条件$\sum_{x\in\{0,1\}}p_j(x|y)=1,\forall y,j$ 有

$$\sum_{x\in\{0,1\}}p_j(x|y)=\frac{\text{count}(y)}{\beta_j(y)}=1,\forall y,j$$

得到$\beta_j(y)=\text{count}(y)$，因此

$$p_j(x|y)=\frac{\text{count}_j(x|y)}{\beta_j(y)}=\frac{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\wedge x_j^{(i)}=x\right)}{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)}$$

- 求解结果

$$p(y)=\frac{\text{count}(y)}{m}=\frac{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)}{m}$$

$$p_j(x|y)=\frac{\text{count}_j(x|y)}{\text{count}(y)}=\frac{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\wedge x_j^{(i)}=x\right)}{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)}$$

4.4.4 Laplace Smoothing 拉普拉斯平滑

$$p(y)=\frac{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)+1}{m+k}$$

$$p_j(x|y)=\frac{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\wedge x_j^{(i)}=x\right)+1}{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)+v_j}$$

其中v_j表示第j个特征的可能取值，如$x_j\in\{0,1\}$，那么$v_j=2$.

4.5 Convex Function 凸函数

4.5.1 凸集合

如果集合C中任意两点之间的线段位于C中，则集合C是凸的，也就是说，$\forall x_1,x_2\in C$，$0\leq\theta\leq1$，我们有

$$\theta x_1+(1-\theta)x_2\in C$$

4.5.2 凸函数

一个函数$f:\mathbb{R}^n\to R$是凸的，如果f的定义域是凸的，对于$x,y\in\text{dom}f$ 和 $0\leq\lambda\leq1$，我们有

$$f(\lambda x+(1-\lambda)y)\leq\lambda f(x)+(1-\lambda)f(y)$$

4.5.3 凸函数的充要条件

- 一阶充要条件

$$f(y)\geq f(x)+\nabla_x f(x)^T(y-x)$$

- 二阶充要条件

$$\nabla_x^2 f(x)\succeq 0$$

4.6 Naive Bayes for Multinomial Distribution 多重朴素贝叶斯

4.6.1 定义

- 训练数据$(x^{(i)},y^{(i)})_{i=1,\ldots,m}$
- - $x^{(i)}$是一个n维向量，$x_j^{(i)}\in\{1,2,\ldots,v\},\quad j=1,\ldots,n$
- - $y^{(i)}\in\{1,\ldots,k\}$

$$
\begin{aligned}
& P\left(X_1=x_1, X_2=x_2, \cdots, X_n=x_n | Y=y\right) \\
=& \prod_{j=1}^{n} P\left(X_j=x_j | Y=y\right) \\
=& \prod_{t=1}^{v} \prod_{j: X_j=t} P\left(X_j=t | Y=y\right) \\
=& \prod_{t=1}^{v} \prod_{j: X_j=t} p(t|y) \\
=& \prod_{t=1}^{v} p(t|y)^{count(t)}
\end{aligned}
$$

其中

$$
\operatorname{count}(t)=\sum_{j=1}^{n} \mathbf{1}\left(x_j=t\right)
$$

4.6.2 多重贝叶斯极大似然函数

$$
\begin{aligned}
\ell(\Omega) &= \log \prod_{i=1}^{m} p\left(x^{(i)}, y^{(i)}\right) \\
&= \log \prod_{i=1}^{m} p\left(x^{(i)} | y^{(i)}\right) p\left(y^{(i)}\right) \\
&= \log \prod_{i=1}^{m} \sum_{y=1}^{k} \mathbf{1}\left(y^{(i)}=y\right) p\left(x^{(i)} | y\right) p(y) \\
&= \sum_{i=1}^{m} \log \left(\sum_{y=1}^{k} \mathbf{1}\left(y^{(i)}=y\right) \left(p\left(x^{(i)} | y\right) p(y)\right) \right) \\
&= \sum_{i=1}^{m} \sum_{y=1}^{k} \mathbf{1}\left(y^{(i)}=y\right) \log \left(p\left(x^{(i)} | y\right) p(y)\right) \\
&= \sum_{i=1}^{m} \sum_{y=1}^{k} \mathbf{1}\left(y^{(i)}=y\right) \log \left(p(y) \prod_{j=1}^{n_i} p\left(x_j^{(i)} | y\right) \right) \\
&= \sum_{i=1}^{m} \sum_{y=1}^{k} \mathbf{1}\left(y^{(i)}=y\right) \log \left(p(y) \prod_{t=1}^{v} p(t|y)^{\operatorname{count}^{(i)}(t)} \right) \\
&= \sum_{i=1}^{m} \sum_{y=1}^{k} \mathbf{1}\left(y^{(i)}=y\right) \left(\log p(y) + \sum_{t=1}^{v} \operatorname{count}^{(i)}(t) p(t|y) \right)
\end{aligned}
$$

其中

$$
\operatorname{count}^{(i)}(t)=\sum_{j=1}^{n_i} \mathbf{1}\left(x_j^{(i)}=t\right)
$$

4.6.3 多重贝叶斯极大似然函数求解

$$
\begin{aligned}
&\max \quad \ell(\Omega)=\sum_{i=1}^{m}\sum_{y=1}^{k}\mathbf{1}\left(y^{(i)}=y\right)\left(\log p(y)+\sum_{t=1}^{v}\operatorname{count}^{(i)}(t)p(t|y)\right)\\
&\text{s.t.} \quad p(y)\geq 0,\forall y=1,\cdots,k\\
&p(t|y)\geq 0,\forall t=1,\cdots,v\forall y=1,\cdots,k\\
&\sum_{y=1}^{k}p(y)=1\\
&\sum_{y=1}^{v}p(t|y)=1,\forall y=1,2,\cdots,k
\end{aligned}
$$

- Solution

$$
\begin{aligned}
p(t|y)&=\frac{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)\operatorname{count}^{(i)}(t)}{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)\sum_{t=1}^{v}\operatorname{coun}t^{(i)}(t)}\\
p(y)&=\frac{\sum_{i=1}^{m}\mathbf{1}\left(y^{(i)}=y\right)}{m}
\end{aligned}
$$

4.7 Jensen's inequality 延森不等式

- 假设f是一个凸函数，那么

$$
f\left(\lambda x_1+(1-\lambda)x_2\right)\leq\lambda f\left(x_1\right)+(1-\lambda)f\left(x_2\right)
$$

其中$\lambda\in[0,1]$

- 延森不等式定义

若$f(x)$是定义在I上的凸函数，若$x_1,x_2,\ldots,x_N\in I$，$\lambda_1,\lambda_2,\ldots,\lambda_N\geq 0$，且$\sum_{i=1}^{N}\lambda_i=1$

$$
f\left(\sum_{i=1}^{N}\lambda_i x_i\right)\leq\sum_{i=1}^{N}\lambda_i f\left(x_i\right)
$$

- 证明(归纳证明)

(1) 当$N=1$时，$\lambda=1$，$f(\lambda x)=\lambda f(x)$

(2) 当$N=2$时，根据凸函数定义，明显

$$
f\left(\lambda x_1+(1-\lambda)x_2\right)\leq\lambda f\left(x_1\right)+(1-\lambda)f\left(x_2\right)
$$

(3) 假设$N=k-1$时，等式成立

$$
f\left(\sum_{i=1}^{k-1}\lambda_i x_i\right)\leq\sum_{i=1}^{k-1}\lambda_i f\left(x_i\right)
$$

(4) 当$N=k$时

$$
\begin{aligned}
f\left(\sum_{i=1}^{k} \lambda_i x_i\right) &= f\left(\sum_{i=1}^{k-1} \lambda_i x_i + \lambda_k x_k\right) \\
&= f\left((1-\lambda_k)\sum_{i=1}^{k-1} \frac{\lambda_i}{1-\lambda_k} x_i + \lambda_k x_k\right) \\
&\leq (1-\lambda_k)\, f\left(\sum_{i=1}^{k-1} \frac{\lambda_i}{1-\lambda_k} x_i\right) + \lambda_k f\left(x_k\right) \\
&\leq (1-\lambda_k)\sum_{i=1}^{k-1} \frac{\lambda_i}{1-\lambda_k} f\left(x_i\right) + \lambda_k f\left(x_k\right) \\
&= \sum_{i=1}^{k-1} \lambda_i f\left(x_i\right) + \lambda_k f\left(x_k\right) \\
&= \sum_{i=1}^{k} \lambda_i f(x_i)
\end{aligned}
$$

Lecture5：Support Vector Machine 支持向量

5.1 Hyperplane 超平面

- 将n维空间分成两个半空间
- 定义一个指向外的法向量$w \in \mathbb{R}^n$
- 假设：超平面通过原点，如果不是
- - 有一个偏向项b
  - $b > 0$意味着沿着w方向平移它($b < 0$沿着相反方向)

5.2 函数间隔和几何间隔

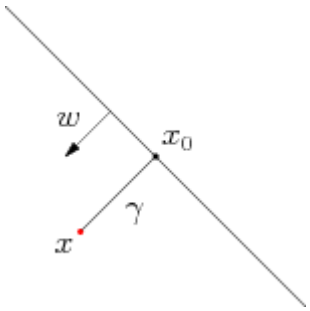
- 函数间隔

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$$

在给定整个训练数据集$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1,\ldots,m}$，函数间隔为

$$\hat{\gamma} = \min_{i=1,\ldots,m} \hat{\gamma}^{(i)}$$

- 几何间隔



1. w为垂直于超平面的一个向量
2. γ为样本x到分类间隔的距离，得到

$$x = x_0 + \gamma \frac{w}{||w||}$$

3. 得到x_0的坐标$(x - \gamma \frac{w}{||w||})$，因此

$$w^T(x - \gamma \frac{w}{||w||}) + b = 0$$

解得

$$\gamma = \frac{w^T x + b}{||w||} = \left(\frac{w}{||w||}\right)^T x^{(i)} + \frac{b}{||w||}$$

4. 这个解只是$y = -1$的情况，所以综合$y = 1$可定义样本x的几何间隔为

$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{w}{||w||}\right)^T x^{(i)} + \frac{b}{||w||} \right)$$

5. 在给定整个训练数据集$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1,\ldots,m}$，几何间隔为

$$\hat{\gamma} = \min_{i=1,\ldots,m} \hat{\gamma}^{(i)}$$

- 几何间隔和函数间隔的关系

$$\text{函数间隔} / ||w|| = \text{几何间隔}$$

5.3 Maximizing The Margin 最大化间隔

最大化几何间隔

$$\hat{\gamma} = \min_{i=1,\ldots,m} \hat{\gamma}^{(i)} = \min_i \left\{ y^{(i)} \left(\left(\frac{\omega}{||w||}\right)^T x^{(i)} + \frac{b}{||\omega||} \right) \right\} = \frac{1}{||\omega||}$$

因此支持向量问题就变为

$$\max_{\omega,b} 1/\|\omega\|$$
$$\text{s.t.} \quad y^{(i)}\left(\omega^T x^{(i)}+b\right)\geq 1, \forall i$$

最大化$1/||w||$就等价于最小化$||w||^2=w^Tw$，即

$$\min_{\omega,b} \omega^T\omega$$
$$\text{s.t.} \quad y^{(i)}\left(\omega^T x^{(i)}+b\right)\geq 1, \forall i$$

5.4 优化问题

- 原始函数

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

- 拉格朗日函数

$$\mathcal{L}(\omega,\alpha,\beta)=f(\omega)+\sum_{i=1}^{k}\alpha_i g_i(\omega)+\sum_{j=1}^{l}\beta_j h_j(\omega)$$

- 拉格朗日对偶函数

$$\begin{aligned}\mathcal{G}(\alpha,\beta)&=\inf_{\omega\in\mathcal{D}}\mathcal{L}(\omega,\alpha,\beta)\\ &=\inf_{w\in\mathcal{D}}\left(f(\omega)+\sum_{i=1}^{k}\alpha_i g_i(\omega)+\sum_{j=1}^{l}\beta_j h_j(\omega)\right)\end{aligned}$$

- 下限属性

为保证最优问题的解大于等于对偶问题的解，因此$\alpha\geq 0$

$$f(\tilde{\omega})\geq\mathcal{L}(\tilde{\omega},\alpha,\beta)\geq\inf_{\omega\in\mathcal{D}}\mathcal{L}(\omega,\alpha,\beta)=\mathcal{G}(\alpha,\beta)$$

- 拉格朗日对偶问题

$$\begin{aligned}\max_{\alpha,\beta} \quad & \mathcal{G}(\alpha,\beta)\\ \text{s.t.} \quad & \alpha_i\succeq 0, \quad \forall i=1,\cdots,k\end{aligned}$$

- KKT条件

为保证原问题的解就是对偶问题的解，假设ω^*和(α^*,β^*)分别是原始问题和对偶问题的解。

$$
\begin{aligned}
&g_i\left(\omega^*\right) \leq 0, \forall i=1, \cdots, k \\
&h_j\left(\omega^*\right)=0, \forall j=1, \cdots, l \\
&\alpha_i^* \geq 0, \forall i=1, \cdots, k \\
&\alpha_i^* g_i\left(\omega^*\right)=0, \forall i=1, \cdots, k \\
&\nabla f\left(\omega^*\right)+\sum_{i=1}^k \alpha_i^* \nabla g_i\left(\omega^*\right)+\sum_{j=1}^l \beta_j^* \nabla h_j\left(\omega^*\right)=0
\end{aligned}
$$

因为ω^*是$\mathcal{L}(\omega, \alpha^*, \beta^*)$的一个极值点，因此$\mathcal{L}(\omega, \alpha^*, \beta^*)$在$\omega^*$处的梯度为0

5.5 凸函数优化问题

- 原始问题

$$
\begin{aligned}
\min_{\omega} \quad & f(w) \\
\text{s.t.} \quad & g_i(w) \leq 0, i=1, \cdots, k \\
& A w-b=0
\end{aligned}
$$

- KKT条件

$$
\begin{aligned}
&g_i(\widetilde{\omega}) \leq 0, \forall i=1, \cdots, k \\
&h_j(\widetilde{\omega})=0, \forall j=1, \cdots, l \\
&\widetilde{\alpha}_i \geq 0, \forall i=1, \cdots, k \\
&\widetilde{\alpha}_i g_i(\widetilde{\omega})=0, \forall i=1, \cdots, k \\
&\nabla f(\widetilde{\omega})+\sum_{i=1}^k \widetilde{\alpha}_i \nabla g_i(\widetilde{\omega})+\sum_{j=1}^l \tilde{\beta}_j \nabla h_j(\widetilde{\omega})=0
\end{aligned}
$$

5.6 支持向量问题求解

- 原始问题

$$
\begin{aligned}
\min_{\omega, b} \quad & \frac{1}{2}\|\omega\|^2 \\
\text{s.t.} \quad & y^{(i)}\left(\omega^T x^{(i)}+b\right) \geq 1, \quad \forall i
\end{aligned}
$$

- 拉格朗日函数

$$
\mathcal{L}(w, b, \alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^m \alpha_i\left(y^{(i)}\left(w^T x^{(i)}+b\right)-1\right)
$$

- 拉格朗日对偶函数

$$
\mathcal{G}(\alpha)=\inf_{w, b} \mathcal{L}(w, b, \alpha)
$$

- 拉格朗日对偶问题

$$
\begin{aligned}
\max_{\alpha} \quad & \mathcal{G}(\alpha) \\
\text{s.t.} \quad & \alpha_i \succeq 0, \quad \forall i=1, \cdots, k
\end{aligned}
$$

- KKT条件求解

$$\nabla_w \mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} = 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\frac{\partial}{\partial b} \mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

拉格朗日对偶问题变形为

$$\begin{aligned} \max_{\alpha} \mathcal{G}(\alpha) =& \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \left(x^{(i)}\right)^T x^{(j)} \\ \text{s.t. } & \alpha_i \geq 0 \quad \forall i \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \end{aligned}$$

一旦我们知道了α^*，那么

$$\omega^* = \sum_{i=1}^{m} \alpha^* y^{(i)} x^{(i)}$$

根据KKT条件，$\alpha_i^* \left(y^{(i)} \left(\omega^{*T} x^{(i)} + b\right) - 1\right) = 0$，因此对于$\forall i$，我们有

$$y^{(i)} \left(\omega^{*T} x^{(i)} + b^*\right) = 1$$

因此$\alpha^* > 0$时，我们有

$$b^* = y^{(i)} - \omega^{*T} x^{(i)}$$

更一般的，我们可以写成

$$b^* = \frac{\sum_{i:\alpha_i^*>0} \left(y^{(i)} - \omega^{*T} x^{(i)}\right)}{\sum_{i=1}^{m} \mathbf{1}\left(\alpha_i^* > 0\right)}$$

5.7 Kernelized SVM

$$\begin{aligned} \max_{\alpha} & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j < x^{(i)}, x^{(j)} > \\ \text{s.t.} & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \\ & \alpha_i \geq 0, \quad \forall i \end{aligned}$$

- Replacing $< x^{(i)}, x^{(j)} >$ by $\phi\left(x^{(i)}\right)^T \phi\left(x^{(j)}\right) = K\left(x^{(i)}, x^{(j)}\right) = K_{ij}$

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K_{i,j} \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \\ & \alpha_i \geq 0, \forall i \end{aligned}$$

- 分割平面的变化

之前

$$y = \text{sign}\left(\omega^T x\right) = \text{sign}\left(\sum_{s\in\mathcal{S}} \alpha_s y^{(s)} {x^{(s)}}^T x\right)$$

核化后

$$y = \text{sign}\left(\sum_{s\in\mathcal{S}} \alpha_s y^{(s)} \phi\left(x^{(s)}\right)^T \phi(x)\right) = \text{sign}\left(\sum_{s\in\mathcal{S}} \alpha_s y^{(s)} K\left(x^{(s)}, x\right)\right)$$

5.8 Soft-Margin SVM 软间隔SVM

在原有的约束条件上加上一个松弛因子ξ，要求$\xi \geq 0$，那么约束条件变为

$$y^{(i)}\left(\omega^T x^{(i)} + b\right) \geq 1 - \xi_i \quad \forall i$$

- 原始问题

$$\begin{array}{ll} \min_{\omega,b,\xi} & \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^m \xi_i \\ \text{s.t.} & y^{(i)}\left(\omega^T x^{(i)} + b\right) \geq 1 - \xi_i, \quad \forall i = 1, \cdots, m \\ & \xi_i \geq 0, \quad \forall i = 1, \cdots, m \end{array}$$

C为惩罚因子。

- 拉格朗日函数

$$\mathcal{L}(\omega, b, \xi, \alpha, r) = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i\left[y^{(i)}\left(\omega^T x^{(i)} + b\right) - 1 + \xi_i\right] - \sum_{i=1}^m r_i\xi_i$$

- KKT条件

$$\begin{array}{l} \nabla_\omega \mathcal{L}(\omega, b, \xi, \alpha, r) = 0 \Rightarrow \omega = \sum_{i=1}^m \alpha_i y^{(i)} x^{(i)} \\ \nabla_b \mathcal{L}(\omega, b, \xi, \alpha, r) = 0 \Rightarrow \sum_{i=1}^m \alpha_i y^{(i)} = 0 \\ \nabla_{\xi_i} \mathcal{L}(\omega, b, \xi, \alpha, r) = 0 \Rightarrow \alpha_i + r_i = C, \text{ for } \forall i \\ \alpha_i, r_i, \xi_i \geq 0, \text{ for } \forall i \\ y^{(i)}\left(\omega^T x^{(i)} + b\right) + \xi_i - 1 \geq 0, \text{ for } \forall i \\ \alpha_i\left(y^{(i)}\left(\omega^T x^{(i)} + b\right) + \xi_i - 1\right) = 0, \text{ for } \forall i \\ r_i\xi_i = 0, \text{ for } \forall i \end{array}$$

- 拉格朗日对偶问题

$$\begin{aligned}\max_{\alpha} \quad & \sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}y^{(i)}y^{(j)}\alpha_i\alpha_j<x^{(i)},x^{(j)}> \\ \text{s.t.} \quad & 0\le\alpha_i\le C, \quad \forall i=1,\cdots,m \\ & \sum_{i=1}^{m}\alpha_i y^{(i)}=0\end{aligned}$$

因为$\alpha_i+r_i=C$，有$r_i=C-\alpha_i, \quad \forall i$

又因为$r_i\xi_i=0$，有$(C-\alpha_i)\xi=0, \quad \forall i$，因此当$\alpha\neq C$时

$$\alpha_i\left(y^{(i)}\left(w^Tx^{(i)}+b\right)-1\right)=0$$

若$0<\alpha_I<C$，有

$$y^{(i)}\left(w^Tx^{(i)}+b\right)=1$$

最终b的解为

$$b=\frac{\sum_{i:0<\alpha_i<C}\left(y^{(i)}-w^Tx^{(i)}\right)}{\sum_{i=1}^{m}1\left(0<\alpha_i<C\right)}$$

- 根据KKT条件推导出来的有用条件

$$\begin{aligned}&\text{When } \alpha_i=0, y^{(i)}\left(w^Tx^{(i)}+b\right)\ge 1\\&\text{When } \alpha_i=C, y^{(i)}\left(w^Tx^{(i)}+b\right)\le 1\\&\text{When } 0<\alpha_i<C, y^{(i)}\left(w^Tx^{(i)}+b\right)=1\end{aligned}$$

5.9 Soft-Margin VS Hard-Margin

C比较大的时候表示我们想要犯更少的错误，但margin会稍微小一点

C比较的小的时候表示我们想要更大的margin，划分错误多一点没关系

lecture6：K-Means

6.1 介绍

- 通常的无监督学习问题
- 给定N个没有标签的样本$\{x_1,\ldots,x_N\}$
- 将示例分组为K个“同类”的分区
- 一个好的聚类是实现
-
  - 群内相似度高
  - 群集间相似度低

6.2 算法步骤

1. 确定类别个数k
2. 随机初始化k个类的中心，分别为$\mu_1, \ldots, \mu_k$
3. 确定每个样本的类别，原则为样本与类中心的距离最小，即

$$\mathcal{C}_{k^*} = \left\{ x_i : k^* = \arg\min_k \|x_i - \mu_k\|^2 \right\}$$

4. 更新每个类的中心

$$\mu_k = \text{mean}(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|} \sum_{x \in \mathcal{C}_k} x$$

5. 若已经收敛，则结束迭代，否则转过程3.迭代是否收敛可以根据本次与前一次每个类的中心变化来确定。

6.3 K-Means 算法优缺点

- 优点：
    1. 理解容易，聚类效果不错
    2. 处理大数据集的时候，该算法可以保证较好的伸缩性和高效率
    3. 当簇近似高斯分布的时候，效果非常不错
- 缺点：
    1. K值是用户给定的，在进行数据处理前，K值是未知的，不同的K值得到的结果也不一样;
    2. 对初始簇中心点是敏感的
    3. 不适合发现非凸形状的簇或者大小差别较大的簇

Lecture7：Principle Component Analysis 主成分分析

7.1 介绍

PCA是一种常用的数据分析方法。PCA通过线性变换将原始数据变换为一组各维度线性无关的表示，可用于提取数据的主要特征分量，常用于高维数据的降维。

- $X = [x^{(1)}\ x^{(2)}\ \ldots\ x^{(N)}]$, $x \in \mathbb{R}^D$, $X \in \mathbb{R}^{D \times N}$
- $Z = [z^{(1)}\ z^{(2)}\ \ldots\ z^{(N)}]$, $z \in \mathbb{R}^K$, $Z \in \mathbb{R}^{K \times N}$
- $u_i \in \mathbb{R}^D$，u_i相当于基向量。

$$Z = \begin{bmatrix} - & u_1^T & - \\ & \vdots & \\ - & u_k^T & - \end{bmatrix} * X$$

7.2 算法流程

1. 特征**标准化**。

$$X = X - \text{mean}(X) = X - \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$$

2. 计算**协方差矩阵**Σ

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}\left(x^{(i)}\right)\left(x^{(i)}\right)^{T} = \frac{1}{N}\cdot XX^{T}$$

3. 求 Σ 的特征向量和特征值：

$$(V, D) = \text{eig}(\Sigma)$$

4. 按照特征值大小重新排序特征向量
5. 取前k个特征向量构成矩阵U

$$U = [v_1\ v_2\ \ldots\ v_k]$$

6. 计算降维后的数据

$$Z = U^{T}X$$

7.3 算法的优缺点

- 优点：

（1）它是无监督学习，完全无参数限制的。在PCA的计算过程中完全不需要人为的设定参数或是根据任何经验模型对计算进行干预，最后的结果只与数据相关，与用户是独立的。

（2）用PCA技术可以对数据进行降维，同时对新求出的“主元”向量的重要性进行排序，根据需要取前面最重要的部分，将后面的维数省去，可以达到降维从而简化模型或是对数据进行压缩的效果。同时最大程度的保持了原有数据的信息。

（3）各主成分之间正交，可消除原始数据成分间的相互影响。

（4）计算方法简单，易于在计算机上实现

- 缺点

（1）如果用户对观测对象有一定的先验知识，掌握了数据的一些特征，却无法通过参数化等方法对处理过程进行干预，可能会得不到预期的效果，效率也不高。

（2）贡献率小的主成分往往可能含有对样本差异的重要信息。

（3）特征值矩阵的正交向量空间是否唯一有待讨论。

（4）在非高斯分布的情况下，PCA方法得出的主元可能并不是最优的，此时在寻找主元时不能将方差作为衡量重要性的标准。