# Contents



本文参考自李航的《统计学习方法》、周志华的《机器学习》等。

# 1 概述

## 1.1 统计学习三要素

### 1.1.1 模型

监督学习中，模型是**要学习的条件概率分布或决策函数**。

#### 1.1.1.1 模型的假设空间

假设空间是所有可能的条件概率分布或决策函数

##### 1.1.1.1.1 定义 1

可以定义为**决策函数的集合**：

$$\mathcal{F} = \{f|Y = f(X)\}$$

- $X$ 和 $Y$ 是定义在 $\mathcal{X}$ 和 $\mathcal{Y}$ 上的变量
- $\mathcal{F}$ 是一个参数向量决定的**函数族**：

$$\mathcal{F} = \{f|Y = f_\theta(X), \theta \in R^n\}$$

参数向量 $\theta$ 取值于 n 维欧式空间 $R^n$，称为**参数空间**

##### 1.1.1.1.2 定义 2

也可以定义为**条件概率的集合**：

$$\mathcal{F} = \{P|P(Y|X)\}$$

- $X$ 和 $Y$ 是定义在 $\mathcal{X}$ 和 $\mathcal{Y}$ 上的**随机变量**
- $\mathcal{F}$ 是一个参数向量决定的**条件概率分布族**：

$$\mathcal{F} = \{P|P_\theta(Y|X), \theta \in R^n\}$$

### 1.1.2 策略

#### 1.1.2.1 损失函数与风险函数

**损失函数（loss function）或代价函数（cost function）**：度量预测值 $f(X)$ 与真实值 $Y$ 的误差程度，记为 $L(Y, f(X))$，是个非负实值函数。损失函数越小，模型越好。

- 0-1 损失函数：

$$L(Y, f(X)) = \begin{cases} 0 & Y \neq f(X) \\ 1 & Y = f(X) \end{cases}$$

- 平方损失函数：

$$L(Y, f(X)) = (Y - f(X))^2$$

- 绝对损失函数：

$$L(Y, f(x)) = |Y - f(X)|$$

- 对数损失函数 (logarithmic loss function)/对数似然损失函数 (log-likelihood loss function):

$$L(Y, P(Y|X)) = -logP(Y|X)$$

**风险函数 (risk function) 或期望损失 (expected loss)**：$X$ 和 $Y$ 服从联合分布 $P(X,Y)$，理论上模型 $f(X)$ 关于**联合分布**$P(X,Y)$ 的平均意义下的损失：

$$R_{exp}(f) = E_P[L(Y, f(X))] = \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x))P(x,y)dxdy$$

学习的目标：选择期望风险最小的模型。但联合分布 $P(X,Y)$ 是未知的，所以无法直接计算 $R_{exp}(f)$。所以监督学习是病态问题（ill-formed problem）：一方面需要联合分布，另一方面联合分布是未知的。

给定训练集：

$$T = \{(x_1, y_1), ...(x_N, y_N)\}$$

**经验风险 (expirical risk)/经验损失 (expirical loss)**：模型 $f(X)$ 关于**训练集**的平均损失

$$R_{emp}(f) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i))$$

根据**大数定律**，当样本容量 $N$ 趋向无穷时，经验风险 $R_{emp}$ 趋于期望风险 $R_{exp}(f)$。

#### 1.1.2.2 经验风险最小化与结构风险最小化

**经验风险最小化（empirical risk minimization, ERM）**：**经验风险最小**的模型就是最优模型。所以需要求解的最优化问题是：

$$\min_{f\in\mathcal{F}} R_{erm} = min_{f\in\mathcal{F}} \frac{1}{N} L(y_i, f(x_i))$$

当满足以下两个条件时，**经验风险最小化**就等价于**极大似然估计**（maximum likelihood estimation）：

- 模型是**条件概率分布**
- 损失函数是**对数损失函数**

当样本量足够大时，ERM 能有很好的效果，但样本量不够多时，为了防止过拟合，需要用下面的方法。

**结构风险最小化（structual risk minimization, SRM）**：结构风险 = 经验风险 + 表示**模型复杂度**的正则化项 (regularizer) 或罚项 (penalty term)。结构风险定义如下：

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

$J(f)$ 是模型的复杂度，模型越复杂，$J(f)$ 越大。$\lambda \geq 0$ 是用于权衡经验风险和模型复杂度的系数。

当满足以下 3 个条件时，**结构化风险最小化**等价于）**贝叶斯估计中的最大后验概率估计** (maximum posterior probability estimation, MAP)：

- 模型是**条件概率分布**
- 损失函数是**对数损失函数**

- 模型复杂度由**模型的先验概率**表示

所以结构风险最小化就是求解优化问题：

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

#### 1.1.3 算法

算法指的是学习模型的具体方法，即使用什么计算方法求解最优模型。

因为统计学习问题归结为最优化问题，所以统计学习的算法就是**求解最优化问题的算法。**

- 如果有显式的解析解，此最优化问题就比较简单
- 如果没有，需要用数值计算方法求解，需要考虑如何**保证找到全局最优解，并使求解过程高效**

### 1.2 模型评估与模型选择

a

#### 1.2.1 训练误差与测试误差

a

#### 1.2.2 过拟合与模型选择

b

### 1.3 正则化与交叉验证

c

#### 1.3.1 正则化

d

#### 1.3.2 交叉验证

e

### 1.4 泛化能力

f

#### 1.4.1 泛化误差

g

#### 1.4.2 泛化误差上界

a

### 1.5 生成模型与判别模型

a

### 1.6 分类问题

a

### 1.7 标注问题

c

### 1.8 回归问题

b

## 2 感知机

d

## 3 k 近邻法

e

## 4 朴素贝叶斯法

x

## 5 决策树

w

## 6 logistic 回归与最大熵模型

o

# 7 支持向量机

u

# 8 提升方法

q

# 9 EM 算法及其推广

e

# 10 隐马尔可夫模型

c

# 11 条件随机场

b

# 12 附录

e

## 12.1 矩阵

e

## 12.2 优化

c

### 12.2.1 拉格朗日乘子法

拉格朗日乘子法 (Lagrange multipliers) 是一种寻找多元函数在**一组约束下**的极值的方法。通过引入拉格朗日乘子，将 $d$ 个变量和 $k$ 个约束条件的最优化问题转化为具有 $d+k$ 个变量的无约束优化问题求解。

#### 12.2.1.1 等式约束

假设 $x$ 是 $d$ 维向量，要寻找 $x$ 的某个取值 $x^*$，使目标函数 $f(x)$ 最小且同时满足 $g(x)=0$ 的约束。

从几何角度看，目标是在由方程 $g(x)=0$ 确定的 $d-1$ 维曲面上，寻找能使目标函数 $f(x)$ 最小化的点。

1. 对于约束曲面 $g(x)=0$ 上的**任意点**$x$，该点的梯度 $\nabla g(x)$ 正交于约束曲面
2. 在**最优点**$x^*$，目标函数 $f(x)$ 在该点的梯度 $\nabla f(x^*)$ 正交于约束曲面

对于**第 1 条**，梯度本身就与曲面的切向量垂直，**是曲面的法向量**，并且**指向数值更高的等值线。**

证明：

参考http://xuxzmail.blog.163.com/blog/static/25131916201032810322765 4/

$z=f(x,y)$ 的等值线：$\Gamma: f(x,y)=c$，两边求微分：

$$df(x,y)=dc$$
$$\frac{\partial f}{\partial x}dx+\frac{\partial f}{\partial y}dy=0$$

看成两个向量的内积：

$$\frac{\partial f}{\partial x}dx+\frac{\partial f}{\partial y}dy=\{\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\}\cdot\{dx,dy\}=0$$

而内积 $a\cdot b=|a||b|cos\theta$ 为 0 说明夹角是 90 度，而 $\{\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\}$ 是梯度向量，$\{dx,dy\}$ 是等值线的切向量，所以梯度向量和切向量是垂直的。

对于**第 2 条**，可以用反证法，如下图，蓝色是 $g(x)=0$，橙色是 $f(x)$ 的等值线 (图里假设 $f(x)=x^2+y^2$)，交点的 $\nabla f(x^*)$ 的梯度和 $g(x)$ 的切面不垂直，那么，可以找到更小的等值线，使夹角更接近 90 度，也就是说，这个点不是真正的最优点 $x^*$。

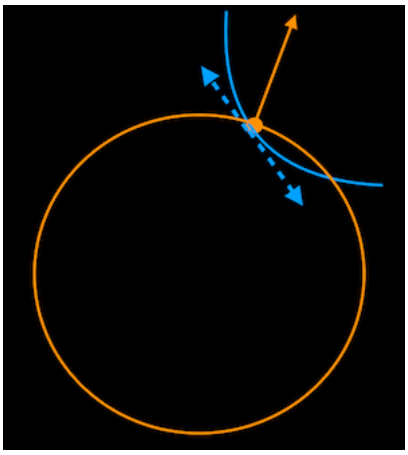

Figure 1: 等式约束-非相切

所以，在最优点 $x^*$ 处，梯度 $\nabla g(x)$ 和 $\nabla f(x)$ 的方向必然**相同或相反**，也就是存在 $\lambda\neq 0$，使得：

$$\nabla f(x^*)+\lambda\nabla g(x^*)=0$$

$\lambda$ 是拉格朗日乘子，定义拉格朗日函数

$$L(x,\lambda)=f(x)+\lambda g(x)$$

$L(x,\lambda)$ 对 x 的偏导 $\nabla_x L(x,\lambda)$ 置 0，就得到：

$$\nabla f(x) + \lambda \nabla g(x) = 0$$

而 $L(x,\lambda)$ 对 $\lambda$ 的偏导 $\nabla_\lambda L(x,\lambda)$ 置 0，就得到

$$g(x) = 0$$

所以，原约束问题可以转化为对 $L(x,\lambda)$ 的无约束优化问题。

#### 12.2.1.2 不等式约束

考虑不等式约束 $g(x) \leq 0$，最优点或者在边界 $g(x) = 0$ 上，或者在区域 $g(x) < 0$ 中。

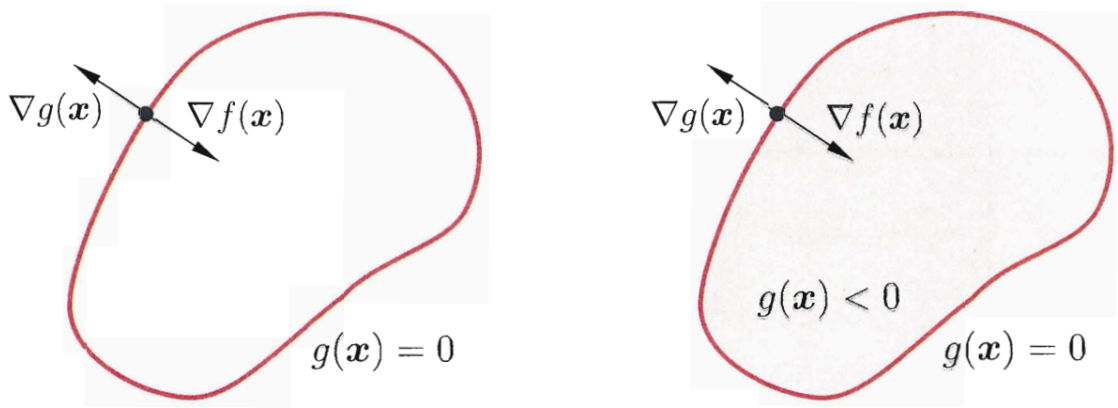


Figure 2: (a) 是等式约束，(b) 是不等式约束

- 对于 $g(x) < 0$

相当于使 $f(x)$ 取得最小值的点落在可行域内，所以**约束条件相当于没有用**，所以，直接对 $f(x)$ 求极小值即可。因为 $L(x,\lambda) = f(x) + \lambda g(x)$，所以

$$\nabla_x L(x,\lambda) = \nabla f(x) + \lambda \nabla g(x)$$

因为 $g(x) < 0$，想要只让 $\nabla f(x) = 0$，那么令 $\lambda = 0$ 即可。

- 对于 $g(x) = 0$

这就变成了等式约束，且此时 $\nabla f(x^*)$ 和 $\nabla g(x^*)$ 反向相反。因为在 $g(x) = 0$ 越往里值是越小的，而梯度是指向等值线高的方向，所以梯度是指向外的。而 $f(x)$ 的可行域又在 $g(x)$ 的里面和边界上，我们要找的是 $f(x)$ 的最小值，所以 $f(x)$ 的梯度是指向内部的。

而 $\nabla f(x) + \lambda \nabla g(x) = 0$，两个又是反向的，所以 $\lambda > 0$。

结合 $g(x) \leq 0$ 和 $g(x) = 0$ 两种情况的结论，就得到了 KKT（Karush-Kuhn-Tucker）条件

$$\left.\begin{matrix} g(x) = 0, \lambda > 0 \\ g(x) < 0, \lambda = 0 \end{matrix}\right\} \Rightarrow \begin{cases} g(x) \leq 0 \\ \lambda \geq 0 \\ \lambda g(x) = 0 \end{cases}$$

其中 $\lambda g(x) = 0$ 是因为 $\lambda$ 和 $g(x)$ **至少一个是 0，而且不能都不是 0**。

以上三个条件有各自的名字：

- Primal feasibility(原始可行性): $g(x) \leq 0$
- Dual feasibility(对偶可行性): $\lambda \geq 0$
- Complementary slackness: $\lambda g(x) = 0$

#### 12.2.1.3 带等式和不等式约束的拉格朗日乘子法

对于多个约束的情形，$m$ 个等式约束和 $n$ 个不等式约束，可行域 $\mathbb{D} \subset \mathbb{R}^d$ 非空的优化问题:

$$\begin{aligned} & \min_x f(x) \\ s.t \quad & h_i(x) = 0, i = 1, ..., m \\ & g_j(x) \leq 0, j = 1, ..., n \end{aligned}$$

引入拉格朗日乘子 $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)^T$ 和 $\mu = (\mu_1, \mu_2, ..., \mu_n)^T$，相应的拉格朗日函数为:

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{n} \mu_j g_j(x)$$

由不等式约束引入的 KKT 条件 $(j = 1, 2, ...n)$ 为

$$\begin{cases} g_j(x) \leq 0 \\ \mu_j \geq 0 \\ \mu_j g_j(x) = 0 \end{cases}$$

### 12.2.2 梯度下降

#### 12.2.2.1 《统计学习方法》的视角

假设 $f(x)$ 有一阶连续偏导，对于无约束的最优化问题而言:

$$\min_{x \in R^n} f(x)$$

$f(x)$ 在 $x^{(k)}$ 附近的一阶泰勒展开如下，其中 $g_k = g(x^{(k)}) = \nabla f(x^{(k)})$ 是 $f(x)$ 在 $x^{(k)}$ 的梯度:

$$f(x) = f(x^{(k)}) + g_k^T (x - x^{(k)})$$

所以对于 $x = x^{(k+1)}$:

$$f(x^{(k+1)}) = f(x^{(k)}) + g_k^T (x^{(k+1)} - x^{(k)})$$

令 $x^{(k+1)} = x^{(k)} + \lambda_k p_k$，$p_k$ 是搜索方向，$\lambda_k$ 是步长，代入上式，有

$$\begin{aligned} f(x^{(k+1)}) &= f(x^{(k)}) + g_k^T (x^{(k)} + \lambda_k p_k - x^{(k)}) \\ &= f(x^{(k)}) + g_k^T \lambda_k p_k \end{aligned}$$

为了让每次迭代的函数值变小，可以取 $p_k = -\nabla f(x^{(k)})$

把 $\lambda_k$ 看成是可变化的，所以需要搜索 $\lambda_k$ 使得

$$f(x^{(k)}+\lambda_k p_k)=\min_{\lambda\geq 0} f(x^{(k)}+\lambda p_k)$$

**梯度下降法**：

输入：目标函数 $f(x)$，梯度 $g(x)=\nabla f(x)$，精度要求 $\varepsilon$。

输出：$f(x)$ 的极小点 $x^*$。

1. 取初始值 $x^{(0)}\in R^n$，置 $k=0$
2. 计算 $f(x^{(k)})$
3. 计算梯度 $g_k=g(x^{(k)})$，当 $\|g_k\|<\varepsilon$，则停止计算，得到近似解 $x^*=x^{(k)}$；否则，令 $p_k=-g(x^{(k)})$，求 $\lambda_k$ 使得

$$f(x^{(k)}+\lambda_k p_k)=\min_{\lambda\geq 0} f(x^{(k)}+\lambda p_k)$$

4. 置 $x^{(k+1)}=x^{(k)}+\lambda_k p_k$，计算 $f(x^{(k+1)})$ 当 $\left\|f(x^{(k+1)})-f(x^{(k)})\right\|<\varepsilon$ 或 $\left\|x^{(k+1)}-x^{(k)}\right\|<\varepsilon$ 时，停止迭代，令 $x^*=x^{(k+1)}$
5. 否则，置 $k=k+1$，转第 3 步

只有当目标函数是**凸函数**时，梯度下降得到的才是**全局最优解**。

#### 12.2.2.2 《机器学习》的视角

梯度下降是一阶 (first order)（只用一阶导，不用高阶导数）优化方法，是求解无约束优化问题最简单、最经典的方法之一。

考虑无约束优化问题 $\min_x f(x)$，$f(x)$ 是连续可微函数，如果能构造一个序列 $x^0,x^1,x^2,...$ 满足

$$f(x^{t+1})<f(x^t),\ t=0,1,2,...$$

那么不断执行这个过程，就可以收敛到局部极小点，根据泰勒展开有：

$$\begin{aligned}f(x)&=f(x^{(k)})+\nabla f(x^{(k)})^T(x-x^{(k)})\\ f(x+\Delta x)&=f(x^{(k)})+\nabla f(x^{(k)})^T(x+\Delta x-x^{(k)})\\ &=f(x^{(k)})+\nabla f(x^{(k)})^T(x-x^{(k)})+\nabla f(x^{(k)})^T\Delta x\\ &=f(x)+\nabla f(x^{(k)})^T\Delta x\end{aligned}$$

而 $\nabla f(x^{(k)})^T\Delta x$ 是一个标量，其转置等于自己，所以

$$f(x+\Delta x)=f(x)+\Delta x^T\nabla f(x^{(k)})$$

想要让 $f(x+\Delta x)<f(x)$，只需要令：

$$\Delta x=-\gamma\nabla f(x)$$

其中的步长 $\gamma$ 是一个小常数

如果 $f(x)$ 满足 $L$-Lipschitz 条件，也就是说对于任意的 $x$，存在常数 $L$，使得 $\|\nabla f(x)\|\leq L$ 成立，那么设置步长为 $\frac{1}{2L}$ 就可以确保收敛到局部极小点。

同样地，当目标函数是凸函数时，局部极小点就对应全局最小点，此时梯度下降可以确保收敛到全局最优解。

### 12.2.3 牛顿法

#### 12.2.3.1 二阶导基本性质

对于点 $x = x_0$，

- 一阶导 $f'(x_0) = 0$ 时，如果二阶导 $f''(x_0) > 0$，那么 $f(x_0)$ 是极小值，$x_0$ 是极小点
- 一阶导 $f'(x_0) = 0$，如果二阶导 $f''(x_0) < 0$，那么 $f(x_0)$ 是极大值，$x_0$ 是极大点
- 一阶导 $f'(x_0) = 0$，如果二阶导 $f''(x_0) = 0$，那么 $x_0$ 是鞍点

证明：

对于任意 $x_1$，根据二阶泰勒展开，有

$$f(x_1) = f(x_0) + f'(x_0)(x_1 - x_0) + \frac{1}{2}f''(x_0)(x_1 - x_0)^2 + ... + R_n(x_1)$$

因为 $f''(x_0) > 0$ 且 $f'(x_0) = 0$，所以，不论 $x_1 > x_0$ 还是 $x_1 < x_0$，总有 $f(x_1) > f(x_0)$，也就是周围的函数值都比 $f(x_0)$ 大，而 $x_0$ 又是极值点，所以是极小点。

#### 12.2.3.2 牛顿法

对于矩阵形式，$x$ 是一个 nx1 的列向量，$H(x)$ 是 $f(x)$ 的海赛矩阵，即二阶导，shape 是 $n \times n$：

$$f(x) = f(x^{(x)}) + g_k^T(x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T H(x^{(k)})(x - x^{(k)})$$

函数 $f(x)$ 有极值的必要条件是在极值点处一阶导为 $0$，特别地，当 $H(x^{(k)})$ 是正定矩阵时（二阶导大于 $0$），是极小值。

牛顿法利用极小点的必要条件 $\nabla f(x) = 0$，每次迭代从点 $x^{(k)}$ 开始，求目标函数极小点，作为第 $k+1$ 次迭代值 $x^{(k+1)}$，具体地，假设 $\nabla f(x^{(k+1)}) = 0$，有

$$\begin{aligned} f(x) &= f(x^{(x)}) + g_k^T(x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T H(x^{(k)})(x - x^{(k)}) \\ &= f(x^{(x)}) + [g_k^T + \frac{1}{2}(x - x^{(k)})^T H(x^{(k)})](x - x^{(k)}) \\ &= f(x^{(x)}) + [g_k + \frac{1}{2}H(x^{(k)})(x - x^{(k)})]^T(x - x^{(k)}) \end{aligned}$$

把其中的 $g_k + \frac{1}{2}H(x^{(k)})(x - x^{(k)})$ 看成一阶导，则上式就是一阶泰勒展开。记 $H^k = H(x^{(k)})$，令 $x = x^{(k+1)}$，令一阶导为 $0$：

$$\begin{gathered} g_k + \tfrac{1}{2}H^k(x^{(k+1)} - x^{(k)}) = 0 \\ g_k = -\tfrac{1}{2}H^k(x^{(k+1)} - x^{(k)}) \\ -2H_k^{-1}g_k = x^{(k+1)} - x^{(k)} \\ x^{(k+1)} = -2H_k^{-1}g_k + x^{(k)} \end{gathered}$$

可以无视这个 $2$，变成：

$$x^{(k+1)} = x^{(k)} - H_k^{-1}g_k$$

或者

$$x^{(k+1)} = x^{(k)} + p_k$$

其中，

$$H_k p_k = -g_k$$

**牛顿法**:

输入：目标函数 $f(x)$，梯度 $g(x) = \nabla f(x)$，海赛矩阵 $H(x)$，精度要求 $\varepsilon$。

输出：$f(x)$ 的极小点 $x^*$。

1. 取初始点 $x^{(0)}$，置 $k = 0$
2. 计算 $g_k = g(x^{(k)})$
3. 若 $\|g_k\| < \varepsilon$，则停止计算，得到近似解 $x^* = x^{(k)}$
4. 计算 $H_k = H(x^{(k)})$，并求 $p_k$，满足
$$H_k p_k = -g_k$$
5. 置 $x^{(k+1)} = x^{(k)} + p_k$
6. 置 $k = k + 1$，转到第 2 步

其中的步骤 4，求 $p_k$ 时，$p_k = -H_k^{-1} g_k$ 需要求解 $H_k^{-1}$ 很复杂。

#### 12.2.4 拟牛顿法的思路

基本想法就是通过一个 $n$ 阶矩阵 $G_k = G(x^{(k)})$ 来近似代替 $H^{-1}(x^{(k)})$。

#### 12.2.5 DFP(Davidon-Fletcher-Powell)

x

#### 12.2.6 BFGS(Broydon-Fletcher-Goldfarb-Shanno)

x

### 12.3 拉格朗日对偶性

x