text(green, it) }

Mathematics Foundation in AI

Contents

1. 第一章: 机器学习与优化建模

1.1. 矩阵奇异值分解

1.1.1. 定义

设 $A \in \mathbb{R}^{m \times n}$, 则 A 的 **奇异值分解 (Singular Value Decomposition, SVD)** 为

$$A = U \Sigma V^T$$

其中 $U \in \mathbb{R}^{m \times m}$ 和 $V \in \mathbb{R}^{n \times n}$ 为正交矩阵, $\Sigma \in \mathbb{R}^{m \times n}$ 为对角矩阵, 其对角线上的元素 $\sigma_1, \sigma_2, ..., \sigma_r$ (其中 $r = \min\{m, n\}$) 为 A 的奇异值, 且满足 $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$.

1.1.2. 分解方法

1. 计算 $A^T A$ 和 AA^T 的特征值和特征向量.
2. 设 $A^T A$ 的特征值为 $\lambda_1, \lambda_2, ..., \lambda_n$ (按降序排列), 则 A 的奇异值为 $\sigma_i = \sqrt{\lambda_i}, i = 1, 2, ..., r$.
3. V 的列向量为 $A^T A$ 的单位特征向量.

1.2. 范数

1.2.1. 向量范数

若实值函数 $\|\cdot\|:\mathbb{R}^n\to\mathbb{R}$ 满足下列条件:

1. 正定性: $\|\boldsymbol{x}\|\geq 0, \forall \boldsymbol{x}\in\mathbb{R}^n. \|x\|=0\Leftrightarrow \boldsymbol{x}=\boldsymbol{0}$.
2. 齐次性: $\|\alpha\boldsymbol{x}\|=|\alpha|\,\|\boldsymbol{x}\|, \forall a\in\mathbb{R}, \boldsymbol{x}\in\mathbb{R}^n$.
3. 三角不等式: $\|\boldsymbol{x}+\boldsymbol{y}\|\leq\|\boldsymbol{x}\|+\|\boldsymbol{y}\|, \forall \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n$.

则 $\|\cdot\|$ 为向量范数

定义 L_p 范数为

$$\|\boldsymbol{x}\|_p=\left[\sum_{j=1}^{n}\|\boldsymbol{x}_j\|^p\right]^{1/p},\quad 1\leq p<\infty$$

特别地, 我们有 $\|\boldsymbol{x}\|_\infty=\max_j\left|x_j\right|$.

1.2.2. 矩阵范数

1.2.2.1. 定义

若实值函数 $\|\cdot\|:\mathbb{R}^{m\times n}\to\mathbb{R}$ 满足下列条件:

1. 正定性: $\|A\|\geq 0, \forall A\in\mathbb{R}^{m\times n}. \|A\|=0\Leftrightarrow A=0$.
2. 齐次性: $\|\alpha A\|=|\alpha|\,\|A\|, \forall a\in\mathbb{R}, A\in\mathbb{R}^{m\times n}$.
3. 三角不等式: $\|A+B\|\leq\|A\|+\|B\|, \forall A,B\in\mathbb{R}^{m\times n}$.

则 $\|\cdot\|$ 为矩阵范数

1.2.2.2. 一些常见的矩阵范数

- Frobenius 范数:

$$\|A\|_F=\left[\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}^2\right]^{1/2}$$

- 核范数 (Nuclear norm):

$$\|A\|_*=\sum_{i=1}^{r}\sigma_i$$

  其中 σ_i 为 A 的奇异值, $r=\min\{m,n\}$. 或者我们也可以定义 σ_i 为 A 的非零奇异值, $r=\operatorname{rank}(A)$.

- 谱范数 (Spectral norm):

$$\| A \|_2 = \sigma_1 = \max_{\boldsymbol{x} \neq 0} \frac{\| A\boldsymbol{x} \|_2}{\| \boldsymbol{x} \|_2}$$

其中 σ_1 为 A 的最大奇异值.

1.2.2.3. 矩阵内积

矩阵 $A, B \in \mathbb{R}^{m \times n}$ 的内积定义为

$$\langle A, B \rangle = \operatorname{Tr}\left(A^T B\right) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij}$$

我们有 Cauchy-Schwarz 不等式:

$$\langle A, B \rangle \leq \| A \|_F \| B \|_F$$

1.3. 酉矩阵和酉不变性

1.3.1. 酉矩阵

设 $U \in \mathbb{C}^{n \times n}$, 如果 U 满足 $U^* U = U U^* = I$, 则称 U 为 **酉矩阵 (Unitary Matrix)**. 这里 U^* 为 U 的共轭转置, 即先对 U 中的每个元素取共轭, 再转置.

特别地, 在实数域上, 酉矩阵称为 **正交矩阵 (Orthogonal Matrix)**, 即 $Q^T Q = Q Q^T = I$.

1.3.2. 酉不变性

矩阵范数 $\| \cdot \|$ 如果满足

$$\| UAV \| = \| A \|, \quad \forall U \in \mathbb{C}^{m \times m}, V \in \mathbb{C}^{n \times n} \quad \text{为酉矩阵}.$$

则称 $\| \cdot \|$ 为 **酉不变 (Unitary Invariant)** 的.

向量的 ℓ_2 范数和矩阵的 Frobenius 范数均为酉不变的.

1.4. 经验风险最小化与期望风险最小化模型

1.4.1. 损失函数

损失函数是针对 **单个** 具体的样本而言的, 用于衡量模型预测值与真实值之间的差异. 损失函数通常记作 $\ell(y, f(\boldsymbol{x}; \boldsymbol{\theta}))$, 其中 y 是样本的真实标签, $f(\boldsymbol{x}; \boldsymbol{\theta})$ 是模型对输入 $\boldsymbol{x}$ 的预测输出, $\boldsymbol{\theta}$ 是模型的参数.

1.4.2. 经验风险

经验风险 (Empirical Risk) 是在给定的训练数据集上计算的平均损失, 用于评估模型在训练数据上的表现. 设训练数据集为 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, 则经验风险定义为

$$R_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i; \boldsymbol{\theta}))$$

1.4.3. 期望风险

期望风险 (Expected Risk) 是在整个数据分布上计算的平均损失, 用于评估模型在未见过的数据上的表现. 设数据分布为 $P(\boldsymbol{x}, y)$, 则期望风险定义为

$$R_{\text{exp}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim P}[\ell(y, f(\boldsymbol{x}; \boldsymbol{\theta}))]$$

期望风险刻画的就是统计意义上的 **母体**.

1.4.4. 结构风险

结构风险 (Structural Risk) 是在经验风险的基础上, 加入正则化项以控制模型复杂度, 从而防止过拟合. 结构风险定义为

$$R_{\text{srm}} = R_{\text{emp}} + \lambda J(\boldsymbol{\theta})$$

这里 $J(\boldsymbol{\theta})$ 衡量模型 (参数) 的复杂度, λ 是正则化参数, 用于平衡经验风险和模型复杂度之间的权重. 通常我们用正则化项来惩罚过于复杂的模型, 以提升模型的泛化能力. 在这种情况下, 监督学习就变成了一个最优化问题

$$\min_{\boldsymbol{\theta}} R_{\text{srm}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \left[\frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i; \boldsymbol{\theta})) + \lambda J(\boldsymbol{\theta})\right]$$

1.4.5. Bayes 风险

如果一个算法 h^* 在全体数据集 $\mathbb{D}$ 上是最好的算法, 则它的期望风险 $R_{\text{exp}}(h^*)$ 称为 **Bayes 风险** (Bayes Risk). 换句话说, Bayes 风险是所有可能的算法中期望风险最小的算法所达到的风险水平.

1.4.6. 近似误差, 估计误差与泛化误差

记 1. $\hat{h}_{\mathcal{H}}$ 是基于有限样本集合 $\mathbb{S}$ 根据经验风险最小从有限算法集合 $\mathcal{H}$ 中选出的最佳算法 (i.e. 经验风险最小). 2. $h^*_{\mathcal{H}}$ 是基于全体数据集 $\mathbb{D}$ 根据期望风险最小从有限算法集合 $\mathcal{H}$ 中选出的最佳算法 (i.e. 期望风险最小). 3. 假设 h^* 的真实 Bayes 风险为 R^*

从一般性考虑, 大范围的最优肯定优于子范围的最优, 即:

$$R^* \leq R_{\exp}(h^*_{\mathcal{H}}) \leq R_{\text{emp}}(\hat{h}_{\mathcal{H}})$$

我们定义 1. **近似误差 (Approximation Error)**: $R_{\exp}(h^*_{\mathcal{H}}) - R^*$, 反映了算法集合 $\mathcal{H}$ 的表达能力.

2. **估计误差 (Estimation Error)**: $R_{\text{emp}}(\hat{h}_{\mathcal{H}}) - R_{\exp}(h^*_{\mathcal{H}})$, 反映了有限样本对算法选择的影响.
3. **泛化误差 (Generalization Error)**: $R_{\text{emp}}(\hat{h}_{\mathcal{H}}) - R^*$, 这里把上面两个限制都加上, 可以看出泛化误差是近似误差和估计误差之和.

1.4.7. 泛化误差限

引入 Hoeffding 不等式, 对于独立同分布的随机变量 $Z_1, Z_2, ..., Z_N$, 且 $Z_i \in [a, b]$, 设

$$P\left(\frac{1}{N}S_N - \mathbb{E}[Z] \geq \epsilon\right) \leq \exp\left(-\frac{2N\epsilon^2}{(b-a)^2}\right)$$

特别地, 当 $Z_i \in [0, 1]$ 时, 有

$$P\left(\frac{1}{N}S_N - \mathbb{E}[Z] \geq \epsilon\right) \leq \exp(-2N\epsilon^2)$$

我们考虑二分类问题, 损失函数为 0-1 损失函数, 即

$$\ell(y, f(\boldsymbol{x}; \boldsymbol{\theta})) = \begin{cases} 0, y = f(\boldsymbol{x}; \boldsymbol{\theta}) \\ 1, y \neq f(\boldsymbol{x}; \boldsymbol{\theta}) \end{cases}$$

设训练数据集 $T = \{(\boldsymbol{u}_i, v_i)\}_{i=1}^N$ 独立同分布采样自总体分布 $P(\boldsymbol{u}, v)$, 其中 $\boldsymbol{u}_i \in \mathbb{R}^n$, $v_i \in \{-1, 1\}$ 设 $\mathcal{H} = \{h_1, h_2, ..., h_k\}$ 为有限假设空间, 且 $\beta = |\mathcal{H}|$ 为函数个数.

将 0-1 损失函数 ℓ 视为随机变量, 由于始终有 $\ell \in [0, 1]$, 因此可以使用 Hoeffding 不等式. 对于 **任意一个函数** h, 由

$$P\big(R_{\text{emp}}(h) - R_{\exp}(h) \geq \epsilon\big) \leq \exp(-2N\epsilon^2)$$

现在我们计算在整个假设空间 $\mathcal{H}$ 中, **存在某个函数** h 使得上式成立的概率.

$$
\begin{aligned}
P(\exists h \in \mathcal{H} : R_{\text{emp}}(h) - R_{exp}(h) \geq \varepsilon) &= P(\bigcup_{h \in \mathcal{H}} \left\{ R_{emp}(h) - R_{exp}(h) \geq \varepsilon \right\}) \\
&\leq \sum_{h \in \mathcal{H}} P(R_{emp}(h) - R_{exp}(h) \geq \varepsilon) \\
&\leq \sum_{h \in \mathcal{H}} \exp(-2n\varepsilon^2) \\
&= \mid \mathcal{H} \mid \exp(-2n\varepsilon^2) \\
&= \beta \exp(-2n\varepsilon^2)
\end{aligned}
$$

我们假设这个满足条件的函数为 $h_{\mathcal{H}}$, 则

$$
P(R_{\text{emp}}(h_{\mathcal{H}}) - R_{\text{exp}}(h_{\mathcal{H}}) < \varepsilon) \leq 1 - \beta \exp(-2n\varepsilon^2)
$$

令 $\delta = \beta \exp(-2n\varepsilon^2)$, $\varepsilon(\delta, \beta, N) = \sqrt{\frac{1}{2N} \ln \frac{\beta}{\delta}}$, 则有

$$
P(R_{\text{emp}}(h_{\mathcal{H}}) - R_{\text{exp}}(h_{\mathcal{H}}) < \varepsilon(\delta, \beta, N)) \geq 1 - \delta
$$

这说明至少有 $1 - \delta$ 的概率, 使得估计误差小于 $\varepsilon(\delta, \beta, N)$. 这就找到了估计误差的上界.

对于泛化误差的上节, 我们令

$$
\hat{h}_{\mathcal{H}} = \arg\min_{h \in \mathcal{H}} R_{\text{emp}}(h)
$$

则有

$$
\begin{aligned}
\parallel R_{\text{emp}}(\hat{h}_{\mathcal{H}}) - R^* \parallel &\leq \underbrace{R_{\text{emp}}(\hat{h}_{\mathcal{H}}) - R_{\text{exp}}(h^*_{\mathcal{H}})}_{\text{估计误差}} + \underbrace{R_{\text{exp}}(h^*_{\mathcal{H}}) - R^*}_{\text{近似误差}} \\
&< \varepsilon(\delta, \beta, N) + \textbf{近似误差}
\end{aligned}
$$

这就是泛化误差界, 它刻画了学习算法的经验风险与期望风险之间偏差和收敛速度.

由此可知: 1. 假设空间的复杂度 β 越大, 估计误差的上界越大, 泛化能力越差. 2. 训练样本数 N 越大, 估计误差的上界越小, 泛化能力越强.

1.5. 过拟合与欠拟合

1.5.1. 过拟合

过拟合 (Overfitting) 是指模型在训练数据上表现良好, 但在未见过的测试数据上表现较差的现象. 过拟合通常发生在模型过于复杂, 参数过多, 或训练数据量不足的情况下.

1.5.2. 欠拟合

欠拟合 (Underfitting) 是指模型在训练数据上和测试数据上都表现不佳的现象.

1.5.3. 模型评估

为了定量考虑这些问题, 往往将数据集进行随机分组, 一部分作为训练集 (Training Set), 用于模型训练; 另一部分作为测试集 (Test Set), 用于选择最合适的模型. 通过比较模型在训练集和测试集上的表现, 可以判断模型是否存在过拟合或欠拟合现象.

设在训练集 T 上, 我们训练后的模型为 $h_T(u)$, 那么该模型对数据 u 的预测输出为 $f(\bar{u}) = \mathbb{E}_T[h_T(u)]$.

设验证集样本的真实值为 v, 由于会有噪声的存在, 样本的标签值可能与真实值有出入, 设标签值为 v_T, 噪声 $v_\epsilon = v - v_T$, 这里假设 $v_\epsilon \in \mathcal{N}(0, \sigma^2)$.

定义

1. **偏差 (Bias)**: $\text{Bias}\ (u) = v_T - f(\bar{u})$, 衡量模型预测值的期望与真实值之间的差异.
2. **方差 (Variance)**: $\text{Var}\ (u) = \mathbb{E}_T\left[(h_T(u) - f(\bar{u}))^2\right]$, 衡量模型预测值在不同训练集上的波动性.
3. **泛化误差 (Generalization Error)**: $\text{Err}\ (u) = \mathbb{E}_T\left[(h_T(u) - v)^2\right]$, 衡量模型在新数据上的表现.

通过推导可以得到

$$\text{Err}\ (u) = \text{Bias}^2(u) + \text{Var}\ (u) + \sigma^2$$

1.6. 最优化问题

1.6.1. 最优化问题的一般形式

最优化问题的一般形式为

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathcal{X} \end{aligned}$$

其中

1. $\boldsymbol{x} \in \mathbb{R}^n$ 为决策变量 (Decision Variable).
2. $f : \mathbb{R}^n \to \mathbb{R}$ 为目标函数 (Objective Function).
3. $\mathcal{X} \subseteq \mathbb{R}^n$ 为可行域 (Feasible Region). 特别地, 当 $\mathcal{X} = \mathbb{R}^n$ 时, 称为无约束最优化问题 (Unconstrained Optimization Problem).
4. 集合 $\mathcal{X}$ 通常可以由约束函数 $c_i : \mathbb{R}^n \to \mathbb{R}$ 来定义, 即

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n : c_i(\boldsymbol{x}) \leq 0, i = 1, 2, ..., m; c_i(\boldsymbol{x}) = 0, i = m+1, m+2, ..., m+l\}$$

5. 在所有满足约束条件的决策变量中, 使目标函数取最小值的决策变量 $\boldsymbol{x}^*$ 称为最优解 (Optimal Solution), 即

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$$

1.6.2. 最优化问题的类型

1. 当目标函数和约束函数均为线性函数时, 问题称为 **线性规划 (Linear Programming, LP)**.
2. 当目标函数和约束函数中至少有一个为非线性函数时, 问题称为 **非线性规划 (Nonlinear Programming, NLP)**.
3. 如果目标函数是二次函数而约束函数是线性函数, 则问题称为 **二次规划 (Quadratic Programming, QP)**.
4. 包含非光滑函数的问题称为 **非光滑优化 (Nonsmooth Optimization)**.
5. 不能直接求导数的问题称为 **无导数优化 (Derivative-free Optimization)**.
6. 变量只能取整数的问题称为 **整数规划 (Integer Programming, IP)**.
7. 在线性约束下极小化关于半正定矩阵的线性函数的问题称为 **半定规划 (Semidefinite Programming, SDP)**.
8. 最优解只有少量非零元素的问题称为 **稀疏优化 (Sparse Optimization)**.
9. 最优解是低秩矩阵的问题称为 **低秩优化 (Low-rank Optimization)**.

1.6.3. 全局最优解和局部最优解

对于可行解 $\bar{x} \in \mathcal{X}$, 定义如下概念:

1. 如果 $f(\bar{x}) \leq f(x)$ 对于所有 $x \in \mathcal{X}$ 成立, 则称 $\bar{x}$ 为 **全局极小解 (Global Minimum)**.

2. 如果存在某个 $x \in \mathcal{X} \cap B(\bar{x}, \epsilon)$ 成立, 其中 $B(\bar{x}, \epsilon) = \{x \in \mathbb{R}^n : \| x - \bar{x} \| < \epsilon\}$, 则称 $\bar{x}$ 为 **局部极小解 (Local Minimum)**.
3. 进一步地, 如果 $f(\bar{x}) < f(x)$ 对于所有 $x \in \mathcal{X} \cap B(\bar{x}, \epsilon)$ 且 $x \neq \bar{x}$ 成立, 则称 $\bar{x}$ 为 **严格局部极小解 (Strict Local Minimum)**.
4. 如果一个点是局部极小解, 但不是严格局部极小解, 则称其为 **非严格局部极小解 (Non-strict Local Minimum)**.

1.6.4. 优化算法的收敛性

对于实际的最优化问题, 我们常使用 **迭代法 (Iterative Method)** 来求解. 设 $\{x^k\}$ 为算法产生的迭代序列, 如果在某种范数 $\| \cdot \|$ 下, 对于某个局部 (或全局) 最优解 x^*, 有 $\lim_{k \to \infty} \| x^k - x^* \| = 0$, 则称迭代序列 $\{x^k\}$ **依点列** 收敛于 x^*, 相应的算法称为是 **依点列收敛到局部 (或全局) 最优解**.

如果从任意的出发点 $x^0 \in \mathcal{X}$ 开始, 迭代序列 $\{x^k\}$ 都依点列收敛于某个局部 (或全局) 最优解 x^*, 则称该算法 **全局依点列收敛到局部 (或全局) 最优解**. 对于 **凸优化** 问题, 因为任意局部最优解也是全局最优解, 所以全局依点列收敛到局部最优解等价于全局依点列收敛到全局最优解.

1.6.5. 算法的渐进收敛速度

设 $\{x^k\}$ 为算法产生的迭代序列, x^* 为某个局部 (或全局) 最优解: 1. 算法 (点列) $\mathcal{Q}$-线性收敛 (Q-linear Convergence): 存在

$$\| x^{k+1} - x^* \| \leq \mu \| x^k - x^* \|$$

2. 算法 (点列) $\mathcal{Q}$-超线性收敛 (Q-superlinear Convergence):

$$\lim_{k \to \infty} \frac{\| x^{k+1} - x^* \|}{\| x^k - x^* \|} = 0$$

3. 算法 (点列) $\mathcal{Q}$-次线性收敛 (Q-sublinear Convergence):

$$\lim_{k \to \infty} \frac{\| x^{k+1} - x^* \|}{\| x^k - x^* \|} = 1$$

这里超线性收敛速度最快, 次线性收敛速度最慢, 分别是线性收敛的两个极端.

4. 算法 (点列) $\mathcal{Q}$-二次收敛 (Q-quadratic Convergence): 存在
5. 算法 (点列) $\mathcal{R}$-线性收敛 (R-linear Convergence): 存在一个 $\mathcal{Q}$-线性收敛到 0 的非负数列 $\{v_k\}$, 使得对于所有 $k \geq k_0$, 有

$$\| x^k - x^* \| \leq v_k$$

类似地可以定义 $\mathcal{R}$-超线性收敛和 $\mathcal{R}$-次线性收敛. 从 $\mathcal{R}$-收敛速度的定义可以看出序列 $\{v_k\}$ 的收敛速度被另一个序列 $\{v_k\}$ 所控制, 当知道 v_k 的形式时, 我们也称算法 (点列) 的收敛速度为 $\mathcal{O}(v_k)$.

1.6.6. 优化算法的收敛准则

在实际应用中, 由于计算资源和时间的限制, 我们通常不会让算法无限迭代下去, 而是设定一个合理的停止准则 (Stopping Criterion), 当满足该准则时, 算法停止迭代并输出当前的解作为最终结果.

对于无约束优化问题, 常用的收敛准则有

$$\frac{f(x^k) - f(x^*)}{\max\{|\ f(x^*)\ |, 1\}} < \epsilon_1, \quad \| \nabla f(x^k) \| < \epsilon_2$$

其中 ϵ_1, ϵ_2 为预设的精度阈值, x^* 为问题的最优解 (如果已知的话). 这个准则结合了目标函数值的变化和梯度的大小, 能够较好地反映算法的收敛情况.

对于有约束优化问题, 还需要考虑约束违反度, 即要求最后得到的点满足

$$\begin{aligned} \max\{c_i(x^k), 0\} &< \epsilon_3, \quad i = 1, 2, ..., m \\ | \ c_i(x^k) \ | &< \epsilon_4, \quad i = m+1, m+2, ..., m+l \end{aligned}$$

其中 ϵ_3, ϵ_4 为预设的精度阈值. 这个准则确保了最终解不仅在目标函数上接近最优, 还满足约束条件.

1.6.7. 最优化的实例

1.6.7.1. 最小二乘法线性回归

给定训练数据集 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, 其中 $\boldsymbol{x}_i \in \mathbb{R}^n$ 为输入特征, $y_i \in \mathbb{R}$ 为目标值. 我们假设模型为线性模型, 即

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{x}$$

其中 $\boldsymbol{\theta} \in \mathbb{R}^n$ 为模型参数. 我们使用均方误差 (MSE) 作为损失函数, 即 $\ell(y, f(\boldsymbol{x}; \boldsymbol{\theta})) = (y - f(\boldsymbol{x}; \boldsymbol{\theta}))^2$. 如果有偏置项, 则可以将输入特征扩展为 $\boldsymbol{x}' = [1, \boldsymbol{x}^T]^T$, 参数扩展为 $\boldsymbol{\theta}' = [\theta_0, \boldsymbol{\theta}^T]^T$, 其中 θ_0 为偏置项.

我们定义 **增广矩阵** $\tilde{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]^T \in \mathbb{R}^{N \times n}$ 和 **目标值向量** $\boldsymbol{y} = [y_1, y_2, ..., y_N]^T \in \mathbb{R}^N$. 则经验风险可以写成矩阵形式

$$R_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{N} \parallel \tilde{X}\boldsymbol{\theta} - \boldsymbol{y} \parallel_2^2$$

最优化问题就为 $\min_{\boldsymbol{\theta}} R_{\text{emp}}(\boldsymbol{\theta})$. 这个问题有解析解

$$\boldsymbol{\theta}^* = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T\boldsymbol{y}$$

1.6.7.2. 岭回归

岭回归 (Ridge Regression) 是在最小二乘法的基础上, 引入 L_2 正则化项以防止过拟合. 岭回归的目标函数为

$$R_{\text{ridge}}(\boldsymbol{\theta}) = \parallel \tilde{X}\boldsymbol{\theta} - \boldsymbol{y} \parallel_2^2 + \lambda \parallel \boldsymbol{\theta} \parallel_2^2$$

$$\boldsymbol{\theta}^* = (\tilde{X}^T\tilde{X} + \lambda I)^{-1}\tilde{X}^T\boldsymbol{y}$$

岭回归的几何解释: 岭回归的最优化问题

$$\min_{\boldsymbol{\theta}} \parallel \tilde{X}\boldsymbol{\theta} - \boldsymbol{y} \parallel_2^2 + \lambda \parallel \boldsymbol{\theta} \parallel_2^2$$

等价于约束最优化问题

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \parallel \tilde{X}\boldsymbol{\theta} - \boldsymbol{y} \parallel_2^2 \\ \text{s.t.} \quad & \parallel \boldsymbol{\theta} \parallel_2^2 \leq t \end{aligned}$$

这个约束条件定义了一个以原点为中心的球体, 岭回归的解 $\boldsymbol{\theta}^*$ 必须位于这个球体内.

Woodbury-Sherman-Morrison 公式: 回顾岭回归的解析解

$$\boldsymbol{\theta}^* = (\tilde{X}^T\tilde{X} + \lambda I)^{-1}\tilde{X}^T\boldsymbol{y}$$

设这里 X 的维度为 $N \times D$, 其中 N 为样本数, D 为特征数. 那么 $(\tilde{X}^T\tilde{X} + \lambda I)$ 的维度为 $D \times D$. 当 D 很大时, 计算其逆矩阵的时间复杂度为 $\mathcal{O}(D^3)$, 这在高维数据下是无法接受的.

这时, Woodbury-Sherman-Morrison 公式就派上用场了. 该公式指出, 对于 2×2 的可逆分块矩阵 $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ 满足 A 和 D 可逆, 则有

$$(A - BD^{-1}C)^{-1}BD^{-1} = A^{-1}B(D - CA^{-1}B)^{-1}$$

令 $A = \lambda I, B = \tilde{X}^T, C = -\tilde{X}, D = I$, 则有

$$(\tilde{X}^T\tilde{X} + \lambda I)^{-1}\tilde{X}^T = \tilde{X}^T(\lambda I + \tilde{X}\tilde{X}^T)^{-1}$$

这里的计算复杂度为 $\mathcal{O}(N^3)$. 所以当 $N < D$ 时, 使用 Woodbury-Sherman-Morrison 公式可以显著降低计算复杂度.

1.6.7.3. 稀疏优化

给定 $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, 且有 $m \ll n$. 我们希望找到一个尽可能稀疏的 $x \in \mathbb{R}^n$, 使得 $Ax = b$.

注意到由于 $m \ll n$, 该线性方程组有无穷多解, 重构出原始信号看似很难. 但是, 这些解当中大部分是不重要的, 真正有用的解是所谓的 **稀疏解**, 即原始信号中大部分元素为零, 只有少数元素非零. 因此, 我们可以将问题转化为

$$
\begin{aligned}
&(\ell_p) : \min \| \ x \ \|_p \\
&\text{s.t.} \quad Ax = b, \quad p = 0, 1, 2, ...
\end{aligned}
$$

当 $p = 0$ 时, $\| \ x \ \|_0$ 表示 x 中非零元素的个数, 该问题称为 ℓ_0 **优化问题**. 该问题是 NP-hard 的, $\| \ x \ \|_0$ 的取值只能为整数, 不能使用常规的最优化方法. 同时需要注意的是, ℓ_1 和 ℓ_2 的解也不一定相同.

LASSO 问题: 考虑带 ℓ_1 正则化的最小二乘问题, 即

$$
\min_x \| \ Ax - b \ \|_2^2 + \lambda \| \ x \ \|_1
$$

该问题称为 LASSO (Least Absolute Shrinkage and Selection Operator) 问题, 该问题可以被看作是 ℓ_1 优化问题的一个变种. 通过调整正则化参数 λ, 可以控制解的稀疏性.

1.6.7.4. 回归分析

考虑线性模型 $b = Ax + \epsilon$, 假设 $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, 则我们可以得到给定 A 和 x 时, 观测值 b 的条件概率分布为

$$
p(b \mid A, x) = \frac{1}{(2\pi)^{m/2} \sigma^m} \exp\left(-\frac{1}{2\sigma^2} \| \ Ax - b \ \|_2^2 \right)
$$

我们希望估计一个最优的 x 使得观测值 b 出现的概率最大, 即最大化对数似然函数

$$
\max_x \log p(b \mid A, x) = \max_x -\frac{1}{2\sigma^2} \| \ Ax - b \ \|_2^2 + \text{const}
$$

可以看到这个最优化问题等价于最小化 $\| Ax - b \|_2^2$, 这就是最小二乘法. 也就是说, **当假设误差是高斯白噪声时, 最小二乘解就是线性回归模型的最大似然解**.

Tikhonov 正则化: 为了平衡数据拟合和模型复杂度, 我们可以引入正则化项, 得到 Tikhonov 正则化问题

$$\min_x \| Ax - b \|_2^2 + \lambda \| x \|_2^2$$

这就类似于最小二乘法中的岭回归. 由于正则项的存在, 该问题的目标函数为强凸函数, 解的性质得到改善

LASSO 问题及其变形 另一方面, 如果希望解 x 是稀疏的, 可以添加 ℓ_1 正则化项, 得到 LASSO 问题

$$\min_x \| Ax - b \|_2^2 + \lambda \| x \|_1$$

也可以考虑问题

$$\min_x \| Ax - b \|_2^2, \quad \text{s.t.} \quad \| x \|_1 \leq t$$

考虑到噪声 ϵ 的存在, 我们可以将约束条件改为 $\| Ax - b \|_2^2 \leq \nu$, 这样就得到了一个等价的优化问题

$$\min_x \| x \|_1, \quad \text{s.t.} \quad \| Ax - b \|_2^2 \leq \nu$$

如果参数 x 具有 **分组稀疏性** (Group Sparsity), 即 x 的分量可分为 G 个组, 每个组内的参数必须同时为零或同时非零, 为此人们提出了 **分组 LASSO 问题**:

$$\min_x \| Ax - b \|_2^2 + \mu \sum_{g=1}^{G} \sqrt{n_g} \| x_{\mathcal{J}_g} \|_2$$

其中 $\mathcal{J}_g$ 是第 g 个组的索引, 且

$$n_g = |\mathcal{J}_g|, \quad \sum_{g=1}^{G} n_g = n$$

这里的正则项也可以被看做是 $\| x_{\mathcal{J}_g} \|_2$ 的 ℓ_1 范数, 也就是对每个组的 ℓ_2 范数求和. 分组 LASSO 问题把稀疏性从单个特征提升到了组的级别上, 但不要求组内的稀疏性.

如果需要同时保证分组以及单个特征的稀疏性, 可以考虑将两种正则项结合起来, 即有稀疏分组 LASSO 模型

$$\min_x \| Ax - b \|_2^2 + \mu \sum_{g=1}^{G} \sqrt{n_g} \| x_{\mathcal{J}_g} \|_2 + \lambda \| x \|_1$$

当特征 x 本身不稀疏但在某种变换下是稀疏的, 则需调整正则项

$$\min_x \| Ax - b \|_2^2 + \lambda \| Fx \|_1$$

特别地, 如果要求 x 相邻元素之间是稀疏的 (i.e. 相邻元素之间的差分稀疏), 则可以取

$$F = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n}$$

逻辑回归 (Logistic Regression): 考虑二分类问题, 设 $y \in \{-1, 1\}$, 输入特征为 $\boldsymbol{x} \in \mathbb{R}^n$. 我们假设输出 y 的条件概率分布为

$$p(y = 1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}}};$$

$$p(y = -1 \mid \boldsymbol{x}) = 1 - p(y = 1 \mid \boldsymbol{x}) = \frac{e^{-\boldsymbol{w}^T \boldsymbol{x}}}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}}}$$

这可以被统一为

$$p(y \mid \boldsymbol{x}) = \frac{1}{1 + e^{-y\boldsymbol{w}^T \boldsymbol{x}}}$$

由此可以写出对数似然函数

$$\ell(\boldsymbol{w}) = \sum_{i=1}^{N} \log p(y_i \mid \boldsymbol{x}_i) = -\sum_{i=1}^{N} \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i})$$

在此基础上加上正则项

$$R_{\text{emp}}(\boldsymbol{w}) = \sum_{i=1}^{N} \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}) + \lambda \| \boldsymbol{w} \|_2^2$$

1.6.7.5. 支持向量机 (SVM)

TODO

1.6.7.6. 低秩矩阵恢复

我们考虑下面的实际问题:

某视频网站提供了约 48 万用户对 1 万 7 千多部电影的上亿条评级数据, 希望对用户的电影评级进行预测, 从而改进用户电影推荐系统, 为每个用户更有针对性地推荐影片. 显然每一个用户不可能看过所有的电影, 每一部电影也不可能收集到全部用户的评级. 电影评级由用户打分 1 星到 5 星表示, 记为取值 $1 \sim 5$ 的整数. 我们将电影评级放在一个矩阵 M 中, M 的每一行表示不同用户, 每一列表示不同电影, M_{ij} 表示用户 i 对电影 j 的评级. 由于每个用户只看过部分电影, 因此矩阵 M 是一个稀疏矩阵, 其中大部分元素为 0.

由于用户对电影的评级受多种因素影响, 如用户的兴趣、电影的类型、导演等, 因此我们可以假设矩阵 M 的秩较低. 我们形式化这个问题为, 令 Σ 是矩阵 M 中所有已知评级元素的下标的集合, 即 $\Sigma = \big\{(i,j) : M_{ij} \neq 0\big\}$, 我们希望找到一个低秩矩阵 X 使得

$$\min_{X \in \mathbb{R}^{m \times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad X_{ij} = M_{ij}, \quad (i,j) \in \Sigma$$

回顾矩阵的核范数定义为

$$\| X \|_* = \sum_{i=1}^{\min(m,n)} \sigma_i(X)$$

其中 $\sigma_i(X)$ 是矩阵 X 的奇异值. 由于核范数是一个凸函数, 因此我们可以将上述问题转化为

$$\min_{X \in \mathbb{R}^{m \times n}} \| X \|_* \quad \text{s.t.} \quad X_{ij} = M_{ij}, \quad (i,j) \in \Sigma$$

考虑到观测可能出现误差, 我们可以给出该问题的二次罚函数形式

$$\min_{X \in \mathbb{R}^{m \times n}} \| X \|_* + \lambda \sum_{(i,j) \in \Sigma} (X_{ij} - M_{ij})^2$$

又考虑到低秩矩阵可以被分解 $X = LR^T$, 其中 $L \in \mathbb{R}^{m \times r}$, $R \in \mathbb{R}^{n \times r}$, $r \ll \min(m,n)$ 是矩阵的秩, 因此我们可以将问题转化为

$$\min_{L\in\mathbb{R}^{m\times r},R\in\mathbb{R}^{n\times r}} \sum_{(i,j)\in\Sigma} (L_i^T R_j - M_{ij})^2 + \alpha \parallel L \parallel_F^2 + \beta \parallel R \parallel_F^2$$

这里 $\parallel L \parallel_F$ 和 $\parallel R \parallel_F$ 分别是矩阵 L 和 R 的 Frobenius 范数, 它的作用是消除 L 和 R 在放缩意义下的不唯一性, α 和 β 是正则化参数.

2. 第二章: 凸集与凸函数

2.1. 凸集的定义

2.1.1. 凸集的几何定义

在 $\mathbb{R}^n$ 中, 对于任意两个点 $x_1, x_2 \in \mathbb{R}^n$, 连接这两点的直线定义为

$$\{\theta x_1 + (1-\theta)x_2, \theta \in \mathbb{R}\}$$

特别地, 当 $\theta \in [0,1]$ 时, 该直线退化为线段.

如果过集合 $\mathcal{C}$ 中任意两点 x_1, x_2 的 **直线** 都包含在 $\mathcal{C}$ 中, 即

$$\theta x_1 + (1-\theta)x_2 \in \mathcal{C}, \quad \forall x_1, x_2 \in \mathcal{C}, \forall \theta \in \mathbb{R}$$

则称集合 $\mathcal{C}$ 是 **仿射集 (Affine Set)**. 如果过集合 $\mathcal{C}$ 中任意两点 x_1, x_2 的 **线段** 都包含在 $\mathcal{C}$ 中, 即

$$\theta x_1 + (1-\theta)x_2 \in \mathcal{C}, \quad \forall x_1, x_2 \in \mathcal{C}, \forall \theta \in [0,1]$$

则称集合 $\mathcal{C}$ 是 **凸集 (Convex Set)**. 仿射集肯定是凸集.

2.1.2. 凸集的性质

1. 如果 $\mathcal{S}$ 是凸集, 则 $k\mathcal{S} = \{kx \mid x \in \mathcal{S}, k \in \mathbb{R}\}$ 也是凸集.
2. 如果 $\mathcal{S}$ 和 $\mathcal{T}$ 都是凸集, 则 $\mathcal{S} + \mathcal{T} = \{x + y \mid x \in \mathcal{S}, y \in \mathcal{T}\}$ 也是凸集.
3. 如果 $\mathcal{S}$ 和 $\mathcal{T}$ 都是凸集, 则 $\mathcal{S} \cap \mathcal{T}$ 也是凸集.
4. 凸集的内部 (Interior) 和闭包 (Closure) 也是凸集. 这里
   - 内部定义为 其中 $B(x,r) = \{y \in \mathbb{R}^n : \parallel y - x \parallel < r\}$.
   - 闭包定义为 $\text{cl}\ (\mathcal{C}) = \mathcal{C} \cup \{x : x \text{ 是 } \mathcal{C} \text{ 的极限点}\}$.

2.1.3. 凸组合和凸包

形如

$$\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k, \quad \theta_i \geq 0, i = 1, 2, ..., k; \quad \sum_{i=1}^{k} \theta_i = 1$$

的点称为点 $x_1, x_2, ..., x_k$ 的 **凸组合 (Convex Combination)**

集合 $\mathcal{S}$ 中所有点的凸组合构成的集合称为 $\mathcal{S}$ 的 **凸包 (Convex Hull)**, 记为 $\text{conv}\ (\mathcal{S})$. 显然, $\text{conv}\ (\mathcal{S})$ 是最小的包含 $\mathcal{S}$ 的凸集.

同时, 我们也有 $\text{conv}\ (\mathcal{S}) \subseteq \mathcal{S}$ 当且仅当 $\mathcal{S}$ 是凸集.

2.1.4. 仿射组合和仿射包

形如

$$\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k, \quad \sum_{i=1}^{k} \theta_i = 1$$

的点称为点 $x_1, x_2, ..., x_k$ 的 **仿射组合 (Affine Combination)**. 仿射组合与凸组合的区别在于, 仿射组合的系数 θ_i 可以为负数.

集合 $\mathcal{S}$ 中所有点的仿射组合构成的集合称为 $\mathcal{S}$ 的 **仿射包 (Affine Hull)**, 记为 $\text{aff}\ (\mathcal{S})$. 显然, $\text{aff}\ (\mathcal{S})$ 是最小的包含 $\mathcal{S}$ 的仿射集.

2.1.5. 锥组合和凸锥

相比于凸组合和仿射组合, 锥组合不要求系数之和为 1. 形如

$$\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k, \quad \theta_i \geq 0, i = 1, 2, ..., k$$

的点称为点 $x_1, x_2, ..., x_k$ 的 **锥组合 (Conical Combination)**.

若集合 $\mathcal{S}$ 中的任意点的锥组合都包含在 $\mathcal{S}$ 中, 则称集合 $\mathcal{S}$ 是 **凸锥 (Convex Cone)**.

2.2. 重要的凸集举例

2.2.1. 超平面和半空间

在 $\mathbb{R}^n$ 中, 设 $a \in \mathbb{R}^n, a \neq 0, b \in \mathbb{R}$, 则集合

$$\mathcal{H} = \left\{x \in \mathbb{R}^n : a^T x = b\right\}$$

称为 **超平面 (Hyperplane)**. 超平面将 $\mathbb{R}^n$ 分成两个部分, 即

$$\mathcal{H}_- = \left\{x \in \mathbb{R}^n : a^T x \leq b\right\}, \quad \mathcal{H}_+ = \left\{x \in \mathbb{R}^n : a^T x \geq b\right\}$$

超平面是仿射集, 也是凸集.

半空间 (Half-space) 是指超平面 $\mathcal{H}$ 的任一侧, 即 $\mathcal{H}_-$ 或 $\mathcal{H}_+$. 半空间是凸集, 但不是仿射集.

2.2.2. 多面体

多面体 (Polyhedron) 是指由有限个线性不等式和线性等式所定义的集合, 即

$$\mathcal{P} = \{x \in \mathbb{R}^n : Ax \leq b, Cx = d\}$$

其中 $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, C \in \mathbb{R}^{l \times n}, d \in \mathbb{R}^l$. $x \leq$ 和 $=$ 分别表示逐元素的比较.

多面体是有限个半空间和超平面的交, 因此由凸集的性质可知, 其为凸集.

2.2.3. 范数球和椭球

范数球 (Norm Ball) 是指形如

$$\mathcal{B} = \{x \in \mathbb{R}^n : \| x - x_c \| \leq r\}$$

$\| \cdot \|$ 为某种范数. 范数球是凸集.

椭球 (Ellipsoid) 是指形如

$$\mathcal{E} = \left\{x \in \mathbb{R}^n : (x - x_c)^T P^{-1} (x - x_c) \leq 1\right\}$$

的集合, 其中 $x_c \in \mathbb{R}^n$ 为椭球心, $P \in \mathbb{R}^{n \times n}$ 为对称正定矩阵. 椭球是凸集.

椭球也可以被表示为

$$\mathcal{E} = \{x_c + Au : \| u \|_2 \leq 1\}$$

其中 $A \in \mathbb{R}^{n \times n}$ 满足 $P = AA^T$. 这里的 A 可以被看做是将单位球映射到椭球的线性变换矩阵.

2.2.4. 范数锥

范数锥 (Norm Cone) 是指形如

$$\mathcal{K} = \left\{(x, t) \in \mathbb{R}^{n+1} : \| x \| \leq t, t \geq 0\right\}$$

的集合, 其中 $\| \cdot \|$ 为某种范数. 范数锥是凸锥. 特别地, 用 ℓ_2 范数定义的范数锥称为 **二次锥**

2.2.5. 特殊矩阵集合和 (半) 正定锥

- **对称矩阵集合**: 设 $\mathcal{S}^n$ 表示所有 $n \times n$ 实对称矩阵的集合, 即

$$\mathcal{S}^n = \left\{X \in \mathbb{R}^{n \times n} : X = X^T\right\}$$

- **半正定矩阵集合**: 设 $\mathcal{S}_+^n$ 表示所有 $n \times n$ 实对称半正定矩阵的集合, 即

$$\mathcal{S}_+^n = \{X \in \mathcal{S}^n : X \succcurlyeq 0\}$$

这里 $X \succcurlyeq 0$ 表示矩阵 X 是半正定的, 即对于任意非零向量 $z \in \mathbb{R}^n$, 有 $z^T Xz \geq 0$. 我们一般称 $\mathcal{S}_+^n$ 为 **半正定锥 (Positive Semidefinite Cone)**, 它是一个凸锥.

- **正定矩阵集合**: 设 $\mathcal{S}_{++}^n$ 表示所有 $n \times n$ 实对称正定矩阵的集合, 即

$$\mathcal{S}_{++}^n = \{X \in \mathcal{S}^n : X \succ 0\}$$

2.3. 保凸的运算

2.3.1. 仿射变换的保凸性

设 $f : \mathbb{R}^n \to \mathbb{R}^m$ 是一个仿射变换, 即 $f(x) = Ax + b$, 其中 $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, 则

- 凸集在 f 下的像是凸集

$$f(\mathcal{C}) = \{f(x) : x \in \mathcal{C}\} \text{ is convex if } \mathcal{C} \text{ is convex}$$

- 凸集在 f 下的原像是凸集

$$f^{-1}(\mathcal{D}) = \{x : f(x) \in \mathcal{D}\} \text{ is convex if } \mathcal{D} \text{ is convex}$$

例子, 对于双曲锥

$$\mathcal{C} = \{(x, t) \in \mathbb{R}^{n+1} : \| x \|_2^2 \leq tu, t \geq 0, u \geq 0\}$$