# Machine Learning God Knows

Fduzjrqlw

## 目录

# 1 机器学习理论

## 1.1 简述机器学习问题.

## 1.2 误差分解公式.

## 1.3 风险 (risk), 经验风险 (empirical risk) 的含义.

## 1.4 过拟合, 欠拟合和误差的关系.

# 2 数据预处理

## 2.1 归一化 (normalization) 和标准化 (standardization), 以及它们的相同点和不同点.

## 2.2 为什么要使用归一化/标准化?

## 2.3 什么时候用归一化, 什么时候用标准化?

## 2.4 一定要归一化吗? 举出一些不需要归一化的例子.

# 3 特征工程

## 3.1 特征抽取的目的.

笼统地讲, 增加模型表达能力. 如给线性模型增加非线性性. 人工特征相当于增加了模型偏好, 降低了 approximate error.

## 3.2 独热编码.

# 4 正则化

## 4.1 做正则化的原因.

## 4.2 模型复杂程度的评价指标.

一般来说, 要具体模型具体分析. 通用的有参数个数, 参数值范围, 人工特征数目, 模型结构.

### 4.3 线性回归需要对偏差项 (bias term) 做正则吗?

### 4.4 限制经验风险最小化 (Constrained ERM) v.s. 惩罚项经验风险最小化 (Penalized ERM) 的等价性.

二者各有利弊. 求解的时候我们自然希望优化问题是无约束的, 方便直接套现成的求解器求解. 解释正则性作用的时候一般选择带限制的, 因为约束问题可以转换成两个集合做交, 从图像上更容易观察出一些性质.

### 4.5 $L_1$ 正则的作用和原因.

$L_1$ 正则使得参数更容易变为 0, 给予了特征稀疏性, 起到了特征选择的作用.

解释方法一: 我们以 Lasso 为例. 设 $F(w) = L(w) + \lambda\|w\|_1$, 则可以把惩罚项经验风险最小化问题转换为限制经验分享最小化问题来考虑, 即在给定 $lambda$ 的情况下, 应该对应一个 $r > 0$,

$$\begin{aligned} \inf_{w} \quad & L(w) \\ s.t. \quad & \|w\|_1 \leq r \end{aligned}$$

考虑一系列的等高线 $\{w|L(w) = c\}$, 在 Lasso 回归问题里对应着椭圆, 而在 SVM 里对应着一条条直线 (以 $d = 2$ 讨论). 原问题的答案就变成了求最小的 $c$, 使得 $c$ 所对应的等高线与 $\{\|w\|_1 \leq r\}$ 所对应的正方形相切. 换个想法, 假设正方形上每个点作为切点, 对应的椭圆中心应该落在何处? 绘制区域可以发现, 顶点对应的面积最大, 因此等高线更容易与正方向相切. 同理, 对于斜率不是正负 1 的直线而言, 也更容易与顶点相切. 类似的, 考虑 $L_2$ 正则,$\{\|w\|_2 \leq r\}$ 对应的是圆, 在顶点相切的概率并不高. (相关图片放在 image/1.png 与 2.png 中)

解释方法二: 对参数的假设,$L_1$ 正则假设参数符合拉普拉斯分布, 而 $L_2$ 正则假设参数符合高斯分布. 观察这两个分布的图像可以看出, 拉普拉斯在 0 处的“点概率” 更高, 因为更容易使得参数变为 0.

解释方法三: 从优化算法的角度考虑. 若使用坐标梯度下降法求解, 在迭代过程中, 会出现关于 $w$ 的某个分量 $w_j$ 的优化问题, 即考虑 $F_j(w_j) = L_j(w_j)+\lambda|w_j|+C_j$. 对于这个问题, 可以写成分段函数的形式, 并且在 $w_j = 0$

的时候二者相同, 因而讨论得到, 在 $w_j = 0$ 处取极小值的条件最为宽松. 若使用坐标拆解法, 即将 $d$ 元变量 $w$ 变成 $2d$ 元变量 $w_1 - w_2$, 且满足 $w_1 \geq 0, w_2 \geq 0$. 则应使用带投影的随机梯度下降法, 在求解的时候如果 $w_{1j} < 0$, 则会自动投影为 0, 因此更容易变成 0.

### 4.6 $L_2$ 正则的作用和原因.

$L_2$ 正则使得参数更为平均, 无关特征虽然会靠近 0, 但是一般不会压迫到 0.

### 4.7 为什么要引入 Elastic Net 中的 $L_1$ 和 $L_2$ 的组合正则.

有时候特征并不是独立无关的, 甚至可能是完全相同的. 对于这类特征,$L_1$ 正则的结果是使得部分参数为 0, 而 $L_2$ 正则则会使得参数与对应的特征成正比. 显而易见,$L_2$ 正则带来的效果会使得模型的鲁棒性更好, 泛化能力强. 同时为了保持 $L_1$ 正则特征筛选的能力. 因此将二者进行线性组合.

## 5 优化算法

### 5.1 GD,SGD 与 mini_batch GD 之间的区别和联系.

### 5.2 什么时候用 SGD?mini_batch GD 中 batch_size 的选择.

讨论的时候要考虑到瓶颈是什么? 是内存空间还是计算时间?

### 5.3 梯度下降算法的收敛率 (convergence rate).

需要考虑一般情况和目标函数满足强凸 (正则性) 充分好的情况. 所谓强凸函数, 设 $f$ 可微, 如果存在 $d > 0$, 使得

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{d}{2}\|y - x\|^2, \forall x, y$$

GD 的收敛性在此情况下分为线性收敛 $O(\frac{1}{\varepsilon})$ 和指数收敛 $O(log(\frac{1}{\varepsilon}))$, 其中 $\varepsilon$ 为指定精度. 由于历史原因, 指数收敛有时候被称为线性收敛.

### 5.4 SGD 的使用技巧.

(1) 对于线性模型, 即 $J(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum l(y_t w x_t)$. 如果样本特征稀疏, 则可以考虑先将样本特征以及模型参数稀疏存储, 之后采用分解的方式进行更新.

### 5.5 SGD 的理论依据.

### 5.6 在线学习的动机和方法.

### 5.7 次梯度和次梯度下降算法.

次梯度方向不一定是下降方向, 比如 $f(x_1, x_2) = |x_1| + 2|x_2|$, 在 $x_1$ 轴上一点, 可以找到其次梯度方向就不是下降方向, 注意等高线是棱形.

## 6 模型

### 6.1 支持向量机 (Supported Vector Machine)

#### 6.1.1 SVM 的损失函数是什么?

有两种理解方式. 一种理解方式是几何间隔 (两个超平面之间的距离) + 惩罚项 (几何意义是越过对应分类的超平面的距离); 第二种理解方式是关于 margin 的 hinge_loss+$L_2$ 正则项.

#### 6.1.2 SVM 的推导过程?

#### 6.1.3 为什么要引入对偶问题? 为什么要引入松弛变量 $\xi$?

#### 6.1.4 Slater 条件是什么, 如何验证 SVM 问题满足 Slater 条件.

#### 6.1.5 叙述 SVM 问题的 KKT 条件?

叙述 KKT 条件, 主要从三个方面.

(1) 主问题和对偶问题的约束条件

(2) 松弛互补条件

(3) 优化问题 $g(\lambda^*) = L(\lambda^*, w^*) = inf_w L(\lambda^*, w)$ 导出的一阶偏导数条件.

### 6.1.6 什么叫核方法?

### 6.1.7 核化 (kernelized) 的好处有哪些?

(1) 增加非线性特征, 使得复杂甚至无限维的特征空间成为可能.

(2) 节约内存和计算成本

### 6.1.8 表示定理以及其作用?

表示定理将最优参数的定义域从特征空间 (可能是无限维空间) 限制到了 $\phi(x_1), \cdots, \phi(x_n)$ 生成的线性子空间中. 同时, 获得了最优参数 $w^*$ 的表示, 即 $w^* = \sum \alpha_i \phi(x_i)$, 这样关于 $w$ 的计算可以直接利用已经预处理出来的内积矩阵 $K$. 问题的时间复杂度至此与特征空间的维数无关, 仅与样本点个数 $O(num\_instances)$ 以及核函数 $k(x, y)$ 的计算复杂度有关.

### 6.1.9 如何验证核函数?

有两种方法. 第一种方法是根据 Mercer 定理, 任取 $x_1, x_2, \cdots, x_n$ 去计算 $K = (K(x_i, x_j))_{i,j}$ 是否是半正定阵; 第二种方法利用核函数的加法, 乘法运算和极限运算等基本性质推导.

### 6.1.10 为什么叫径向基函数?

所谓径向基函数, 将第二个变量看成参数的话,$K(x, x_0) = e^{-\gamma \|x - x_0\|_2^2}$, 可以看到函数值与点 $x$ 到 $x_0$ 的距离有关, 即等值面应该是球面, 故向量 $\overrightarrow{x_0 x}$ 对应着球的一条径向.

### 6.1.11 RBF 核对应特征空间的维数.

由 Mercer 定理,RBF 核对应了一个再生核 Hilbert 空间 H 以及 $\phi$, 使得 $K(x, y) =< \phi(x), \phi(y) >_H$. 对于一维的情形, 可以把空间 H 和映射 $\phi$ 写出来.

$$H = l^2 \ , \phi(x) = (e^{\frac{-x^2}{2}} x, \cdots, \frac{e^{\frac{-x^2}{2}}}{\sqrt{i!}} x^i, \cdots)$$

#### 6.1.12 正则化 RBF 核.

正则化 RBF 核其实是对参数 $\alpha_1, \alpha_2, \cdots, \alpha_n$ 的 $L_2$ 范数之和有要求. 更多地用于 RBF 神经网络中, 形式更为明显一些.

#### 6.1.13 参数 $\gamma$ 以及 $C$ 的作用

直观上来看, 参数 $\gamma$ 表明单个样本点的影响大小. 与径向基函数的影响半径成反比. 如果 $\gamma$ 过大, 意味着影响半径很小, 支持向量的影响半径小到只能影响到自己, 因此会出现过拟合的现象. 如果 $\gamma$ 过小, 则任意支持向量的影响半径覆盖到全空间.

参数 $C$ 在误分类和平面光滑性之间做权衡. 如果 $C$ 比较小, 则更允许模型犯错, 平面更为光滑. 如果 $C$ 比较大, 则会牺牲模型光滑性以增加支持向量个数从而减小预测误差.

#### 6.1.14 叙述 SVM 问题的 KKT 条件?

#### 6.1.15 什么是 SVM 的退化?

SVM 的退化是指在求解过程中, 求出的最优参数 $w^* = 0$, 对应的 $a_i^* = 0$ 或 $\frac{C}{n}$. SVM 退化的原因可能是原问题在当前核函数下线形不可分且样本点的分布使得 SVM 的解对应到只和偏置项 $b$ 有关.

#### 6.1.16 如何构造新的输入数据, 使得一个线形可分的 SVM 问题退化?

线形核 SVM 退化当且仅当 (不妨假设 $|K_-| \leq |K_+|$) 存在 $\omega_1, \omega_2, \cdots, \omega_{K_+}, 0 \leq \omega_i \leq 1, s.t. \sum_{i \in K_-} x_i = \sum_{i \in K_+} \omega_i x_i, \sum_{i \in K_+} \omega_i = |K_-|$

### 6.2 感知机 (perceotron)

#### 6.2.1 感知机的工作原理?

#### 6.2.2 感知机与 SVM 的异同点有哪些?

损失函数不同. 感知机的损失函数为

$$\sum \max(0, -m_i) = \sum \max(0, -y_i f(x_i)) = \sum \max(0, -y_i(w^T x_i + b))$$

没有正则化项. 而 SVM 有关于参数 $w$ 的正则化项, 并且关于 margin 的函数为 $\max(0, 1 - m)$.

### 6.2.3 感知机如何优化?

次梯度下降法, 次梯度的找法和 SVM 相同.

### 6.2.4 感知机解的性质

解不唯一, 可能有无穷多组解. 次梯度下降算法产生的解是是所有样本点 $x_i$ 的线性组合, 与 SVM 类似.