CS420 Machine Learning

March 2020
Author: Fu Lingyue
Mail: fulingyue@sjtu.edu.cn

目录

1 Prequisitions

1.1 ML's Goal

Machine learning has a loss function $\mathcal{L}$, and we have to let it take the minimum by learning the parameters of f_θ.

$$\min\{\frac{1}{N}\Sigma_{i=1}^{N}\mathcal{L}(y_i, f_\theta(x_i))\}$$

是预测结果接近真实的标签，是机器学习总体的目标。

1.2 Outline of Proper Nouns

欠拟合、过拟合 算法无法捕捉数据基础变化趋势时，出现欠拟合；模型把随机误差和噪声也考虑进去时，出现过拟合。

正则化 (Regularization) Add a parameter($\lambda\Omega(\theta)$) penalty to prevent the model from overfitting the data. Also in matrix form, it is to prevent singular matrices.

L2 regularization(Ridge): $\Omega(\theta) = ||\theta||_2^2 = \Sigma_{m=1}^{M}\theta_m^2$.

L1 regularization(Lasso): $\Omega(\theta) = ||\theta||_1 = \Sigma_{m=1}^{M}|\theta_m|$.

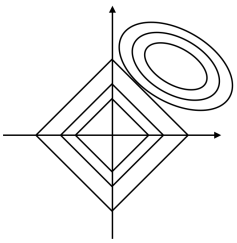

图 1: L1 Reguraization

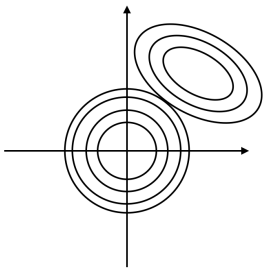

图 2: L2 Reguraization

交叉验证 (Cross Validation) The training data were randomly divided into k groups, and every time we use one group to verify our model.

模型泛化性 (Model Generalization)

判别模型和生成模型 判别模型关注数据中的一维（显式函数）；生成模型关注数据之间的联系（隐函数）.

超参数 模型运行前就根据经验选定的参数，它们只能够人工调整而不能通过学习得到，它们的选取会关系到模型学习的效率、结果、方向等等。

1.3 分类指标

有一类模型，我们用概率 $p_\theta(y|x)$ 来描述它的预测值，叫**概率判别模型**。而预测得到的概率，我们可以设置阈值来分类，包括二分类和多分类。

2 线性判别模型 (Linear Model)

线性回归 (Linear Regression)

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

（本节以**平方误差** $\mathcal{L}(y_i, f_\theta(x_i)) = \frac{1}{2}(y_i - f_\theta(x_i))^2$ 为例）

梯度下降方法 (Gradient Descent Algorithm) 众所周知，梯度的方向是函数下降最快的方向。因此我们对于目标函数关于 θ 取梯度，就可以得到它在 θ 维度下降最快的方向，然后对 θ 进行新的赋值：

$$\theta_{\text{new}} \leftarrow \theta_{\text{old}} - \eta \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

这里的 η 是一个超参数，代表了梯度下降的步长。步长过长会导致越过最优点来回震荡，步长太短会导致收敛速度较慢。

2.1 梯度更新方法

1. 批量梯度下降: 根据整个数据的梯度更新, 准确但是慢；
2. 随机梯度下降：根据某个数据的梯度更新，快但不确定性多；
3. 小批量梯度下降：用 minibatch 的梯度更新，较快，较准，易并行。

Lemma 1. 凸优化目标函数具有唯一最小点。

Lemma1 应用在机器学习中，就意味着一个凸优化目标函数可以从不同的初始化参数最终学习到相同的最优值。其中，(下) 凸函数 $f : \mathbb{R}^n \to \mathbb{R}$ 的定义为：**dom** f 是凸集，并且

$$f(tx_1 + (1-t)x_2) \le tf(x_1) + (1-t)f(x_2) \quad \forall x_1, x_2 \in \mathbf{dom} f, 0 \le t \le 1.$$

2.2 矩阵形式 (Matrix Form)

因为数据集很大，所以我们读入的时候很可能是一个矩阵，因此需要考虑矩阵形式。同时，矩阵形式也能够从另一个角度解释正则化。

训练数据矩阵

$$\text{特征值}\mathbf{X} = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \\ \dots \\ \mathbf{x}^n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ & \dots & & \\ x_1^n & x_2^n & \dots & x_d^n \end{bmatrix}$$

参数 $\mu = (\mu_1 \quad \mu_2 \dots \mu_d)^T$, 标签 $\mathbf{y} = (y_1 \quad y_2 \dots y_n)^T$.

预测结果 $\widehat{\mathbf{y}} = \mathbf{X}\mu = (\mathbf{x^1}\mu \quad \mathbf{x^2}\mu \dots \mathbf{x^n}\mu)$

目标函数 (Loss Function) $J(\mu) = \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{y} - \widehat{\mathbf{y}})^{\mathbf{T}}(\mathbf{y} - \widehat{\mathbf{y}}) = \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{y} - \mathbf{X}\mu)^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\mu)$

为了取到目标函数的最小值，我们需要求梯度为 0 的 μ 值，即

$$\begin{aligned} \frac{\partial J(\mu)}{\partial \mu} &= \frac{1}{2}\frac{\partial(\mathbf{y}^T\mathbf{y} - \mu^T\mathbf{X}\mathbf{y} - \mathbf{y}\mathbf{X}\mu + \mu^T\mathbf{X}^T\mathbf{X}\mu)}{\partial \mu} \\ &= \frac{1}{2}\frac{\partial(\mathbf{y}^T\mathbf{y} - 2\mu^T\mathbf{X}\mathbf{y} + \mu^T\mathbf{X}^T\mathbf{X}\mu)}{\partial \mu} \\ &= \frac{1}{2}\frac{\partial(-2\mu^T\mathbf{X}\mathbf{y} + \mu^T\mathbf{X}^T\mathbf{X}\mu)}{\partial \mu} (\text{J 是数字}) \\ &= -\mathbf{X}\mathbf{y} - ((\mathbf{X}^T\mathbf{X}) + (\mathbf{X}^T\mathbf{X})^T)\mu \\ &= -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mu) = 0 \end{aligned}$$

这里用到了矩阵求导法则 (See Appendix A). 接着求解方程：

$$-\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mu) = 0$$

$$\mathbf{X^T}\mathbf{X}\mu = \mathbf{X^T}\mathbf{y}$$

$$\mu = (\mathbf{X^T}\mathbf{X})^{-\mathbf{1}}\mathbf{X^T}\mathbf{y}$$

至此我们得到了最优的 $\mu = (\mathbf{X^T}\mathbf{X})^{-\mathbf{1}}\mathbf{X^T}\mathbf{y}$，注意这里的 $\mathbf{X}$ 和 $\mathbf{X}^T$ 不一定有逆矩阵，所以不能化简消去。同时，如果 $\mathbf{X^T}\mathbf{X}$ 是奇异矩阵，那么连 $\mathbf{X^T}\mathbf{X}$ 的逆矩阵都无法计算。因此，我们引入了正则化（正则化的第二个用处）：

$$J(\mu) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mu)^T(\mathbf{y} - \mathbf{X}\mu) + \frac{\lambda}{2}||\mu||_2^2$$

相似的可以得到梯度 $\frac{J(\mu)}{\mu} = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mu) + \lambda\mu$, 解出最优参数 $\hat{\mu} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$.

2.3 泛线性模型

泛线性模型是指，将给的特征映射到新的域上去，再用新的矩阵作为特征数据进行学习。它只是用了一个特征矩阵 $\mathbf{n*h}$（或者说是特征映射函数 $\phi(x) : \mathbb{R}^d \mapsto \mathbb{R}^h$），进行了 x 到 $\phi(x)$ 的转化。

类似的，我们有目标函数 $J(\theta) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\,\theta)^T(\mathbf{y} - \mathbf{X}\,\theta)$，梯度 $\frac{\partial J(\theta)}{\partial \theta} = -\,^T(\mathbf{y} - \mathbf{X}\theta)$ 以及最优参数以及最优参 $\hat{\theta} = (\,^{\mathbf{T}}\,)^{-\mathbf{1}}\,^{\mathbf{T}}\mathbf{y}$.

引入 $L2$ 正则化时，需要利用线性技巧来得到最优参数和预测值（略）。这里的 K 也被称为核矩阵，所以也叫**核线性回归**。

2.4 逻辑回归 (Logistic Regression)

我们这里讨论的是概率判别模型，也就是用概率 $p_\theta(y|x)$ 来描述某一类别的可能性。

2.4.1 二分类

概率特点 只有 $y = 0$ 和 $y = 1$ 两种可能性，就有如下关系：

$$p_\theta(y = 0|x) = 1 - p_\theta(y = 1|x)$$

而概率的大小则由一个 sigmoid 函数来决定，$\sigma(x) = \frac{1}{1 + e^{-x}}$.

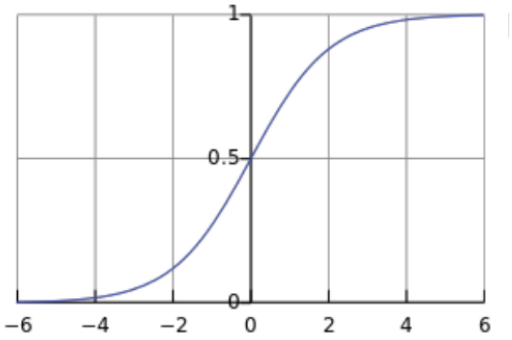

图 3: sigmoid 函数

Sigmoid 函数将一个数字映射到 $(0, 1)$ 区间里，我们把它看作是选择概率。

$$p_\theta(y = 1|x) = \sigma(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$p_\theta(y = 0|x) = 1 - \sigma(\theta^T x) = \frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}}$$

选择 sigmoid 函数的原因是它不仅能够映射到 (0,1), 而且它有一个很好的性质:

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1-\sigma(z))$$

损失函数——交叉熵函数

$$H(p,q) = -\Sigma_x p(x)\log q(x)$$

$$H(p,q) = -\int_x \log q(x)\mathrm{d}x$$

把二分类的概率特点代入后，得到

$$\mathcal{L}(y,x,p_\theta) = -y\log p_\theta(y=1|x) - (1-y)\log(1-p_\theta(y=1|x))$$

这就是二分类所使用的损失函数了。

梯度 依然使用梯度下降法，通过求损失函数对于 θ 求偏导来得到梯度方向:

$$\begin{aligned}\frac{\partial \mathcal{L}(y,x,p_\theta)}{\partial \theta} &= -y\frac{1}{\sigma(\theta^T x)}\sigma(\theta^T x)(1-\sigma(\theta^T x))x - (1-y)\frac{-1}{1-\sigma(\theta^T x)}\sigma(\theta^T x)(1-\sigma(\theta^T x))x \\ &= -y(1-\sigma(\theta^T x))x + (1-y)\sigma(\theta^T x)x \\ &= (\sigma(\theta^T x) - y)x\end{aligned}$$

得到梯度后，更新 $\theta = \theta + \lambda\frac{\partial \mathcal{L}}{\partial \theta}$

预测结果 通过选择概率更大的标签，得到新的预测结果:

$$\hat{y} = \begin{cases} 1, & p_\theta(y=1|x) > h(=0.5) \\ 0, & \text{otherwise} \end{cases}$$

迭代 再用预测结果去和实际的标签比较，得到新的 loss 函数，再次求梯度，再次更新参数 $\theta\cdots$

附录 B 里展示了这部分的函数代码。

2.4.2 多分类

预测 $y = c_j$ 概率

$$p_\theta(y=c_j|x) = \frac{e^{\theta_j^T x}}{\Sigma_{k=1}^m e^{\theta_k^T x}} \quad \text{for} j = 1,2,\ldots,m$$

在这里，Sigmoid 函数变化成了 Softmax 函数，将预测结果 $\{\theta_k^T x\}$ 标准化.

多分类交叉熵

$$\mathcal{L}(y, x, p_\theta) = -\Sigma_k \delta(y = c_k) \log p_\theta(y = c_k|x)$$

它的梯度为

$$\begin{aligned}
\frac{\partial \log p_\theta(y = c_j|x)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \log \frac{e^{\theta_j^T x}}{\Sigma_{k=1}^m e^{\theta_k^T x}} \\
&= x - \frac{\partial}{\partial \theta_j} \log \Sigma_{k=1}^m e^{\theta_k^T x} \\
&= x - \frac{e^{\theta_j^T x}}{\Sigma_{k=1}^m e^{\theta_k^T x}} x \\
&= (1 - p_\theta(y = c_j|x))x
\end{aligned}$$

A 矩阵求导法则

1. $\partial(XY) = (\partial X)Y + X\partial Y$;
2. $\partial(X^T) = (\partial X)^T$;
3. $\partial X^{-1} = -X^{-1}(\partial X)X^{-1}$;
4. $\partial X^T = (\partial X)^T$;
5. $\frac{\partial x^T b}{\partial x} = \frac{\partial b^T x}{\partial x} = b$;
6. $\frac{\partial a^T X b}{\partial X} = ab^T$;
7. $\frac{\partial a^T X a}{\partial X} = \frac{\partial a^T X^T a}{\partial X} = aa^T$;
8. $\frac{\partial b^T X^T X c}{\partial X} = X(bc^T + cb^T)$;
9. $\frac{\partial x^T B x}{\partial x} = (B + B^T)x$.

B 逻辑回归代码块

B.1 二分类

```
def sigmoid(x):
      return 1.0 / (1 + np.exp(-x))
```

```
def l2_loss(self, X, Y, reg):
  mul = np.matmul(X, self.W)
  predict = sigmoid(mul)
  error =  predict - Y
  loss = - np.matmul(Y.T ,np.log(predict))
         - np.matmul((1 - Y).T, np.log(1-predict))
         + reg * np.matmul(W.T,W)/2
  grad = np.matmul(X.T, error) + reg * self.W
  return loss, grad
```

```
def predict(self, X, threshold=0.5):
    mul = np.matmul(X, self.W)
    pre = sigmoid(mul)
    Y_pred = []
    for i in range(X.shape[0]):
        Y_pred.append(mul[i] > threshold)
    return Y_pred
```

B.2 多分类