线性模型

线性模型(Linear Model)是机器学习中应用最广泛的模型，指通过样本特征的线性组合来进行预测的模型。给定一个D维的样本特征的线性组合来进行预测的模型，给定一个D维样本$x=[x_1,x_2,\dots,x_D]^\top$，其线性组合函数为：

$$\begin{aligned} f(x;w) &= w_1x_1+w_2x_2+\cdots+w_Dx_D+b \\ &= w^\top x+b \end{aligned}$$

其中$w=[w_1,\dots,w_D]^\top$为D维的权重向量，b为偏置。上式子可以用于用于线性回归模型：$y=f(x;w)$，其输出为连续值，因此适用于回归预测任务。但是在分类任务中，由于输出目标是离散标签，无法直接进行预测，此时一般引入一个非线性的决策函数$g(\cdot)$来预测输出目标：

$$y=g\circ f(x;w)$$

$f(x;w)$又称为判别函数。

如果$g(\cdot)$的作用是将f函数值挤压/映射到某一值域内，那么$g(\cdot)$称为激活函数。

1.线性回归

这个属于很基础的模型了，它的任务很简单，就是预测连续的标签。

对于给定的样本ξ，我们可以用m个x_i表示其特征，那么可以将原始样本映射称为一个m元的特征向量$\mathbf{x}=(x_1,x_2,\dots,x_m)^T$。因此，我们可以将线性回归模型的初始模型表示为如下的线性组合形式：

$$f(\mathbf{x};\mathbf{w})=w_1x_1+w_2x_2+\cdots+w_mx_m$$

其中，$\mathbf{w}=(w_1,w_2,\dots,w_m)^T$为参数向量。

参数学习方法

定义损失函数为平方误差损失函数：

$$\mathcal{R}(w)=\sum_{i=1}^n [y_i-f(x_i)]^2$$

令训练样本集的特征矩阵为$X=(x_1,x_2,\dots,x_n)=(x_{ij})_{m\times n}$。相应的训练样本标签值为$y=(y_1,y_2,\dots,y_n)^T$，可将上述损失函数转化为：

$$\mathcal{R}(w)=(y-X^\top w)^\top(y-X^\top w)$$

因此，线性回归模型的构造就转化为如下最优化问题：

$$\arg\min_w \mathcal{R}(w)=\arg\min_w (y-X^\top w)^\top(y-X^\top w)$$

$\mathcal{R}(w)$对参数向量w各分量求偏导数：

$$
\begin{aligned}
\frac{\partial \mathcal{R}(w)}{\partial w} &= \frac{\partial (y - X^\top w)^\top (y - X^\top w)}{\partial w} \\
&= \frac{\partial (y^\top - w^\top X)(y - X^\top w)}{\partial w} \\
&= \frac{\partial (y^\top y - y^\top X^\top w - w^\top X y + w^\top X X^\top w)}{\partial w} \\
&= -\frac{\partial y^\top X^\top w}{\partial w} - \frac{\partial w^\top X y}{\partial w} + \frac{w^\top X X^\top w}{\partial w} \\
&= -Xy - Xy + (XX^\top + XX^\top) w \\
&= -2Xy + 2XX^\top w \\
&= 2X(X^\top w - y)
\end{aligned}
$$

根据多元函数求极值的方式，我们令$\mathcal{R}(w)$对参数向量w各分量的偏导数为0，即：

$$
\frac{\partial \mathcal{R}(w)}{\partial w} = 2X(X^\top w - y) = 0
$$

展开，移项，可得:

$$
w = (XX^\top)^{-1} X y
$$

这便是直接利用最小二乘法求解线性回归模型的式子。可以发现里面涉及到了矩阵求逆的操作，这使得最小二乘法自带了明显的限制性：要求X的行向量之间线性无关，即不同样本的属性标记值之间不能存在线性相关性。

但实际应用中大多数样本中都存在这个问题，所以常用另一种方法来优化参数：梯度下降法。

梯度下降算法可以用于求解多元函数极值问题，具体来说，对于函数$f(w)$，设其在某点的梯度为$grad\ f(w) = \bigtriangledown f(w)$，为一矢量，则$f(w)$方向导数沿该方向取得最大值，即$f(w)$沿该方向变化最快(增大)。那么在该点沿梯度负方向减小最快。我们可以从该点沿梯度方向下降一小段(即为η，实际上我们称之为步长/学习率)，到达下一个点，再沿新店的梯度反方向继续下降，如此往复求得函数极值：

$$
w_{i+1} = w_i - \eta \frac{\partial f(w)}{\partial w_i}
$$

以上便是线性回归常用的参数学习方法。

2.Logistic回归

Logistic回归用于解决二分类问题，而不是回归问题。

回到线性分类模型：

$$
y = g \circ f(x; w)
$$

$g(\cdot)$函数在此处的作用是激活函数，用于对函数值进行映射。在$Logistic$回归中，使用$Sigmoid$函数作为激活函数：

$$
\sigma(x) = \frac{1}{1 + e^{-x}}
$$

其对x的导数为：

$$
\begin{aligned}
\frac{\mathrm{d}\sigma(x)}{\mathrm{d}x} &= \frac{\mathrm{d}(1+e^{-x})^{-1}}{\mathrm{d}x} \\
&= -(1+e^{-x})^{-2} \times (-e^{-x}) \\
&= \frac{1}{1+e^{-x}} \times \frac{e^{-x}}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \times (1 - \frac{1}{1+e^{-x}}) \\
&= \sigma(x)(1-\sigma(x))
\end{aligned}
$$

在二分类问题中，我们假设标签取$\{0,1\}$，则标签$y=1$的后验概率为：

$$
p(y=1|x) = \sigma(w^\top x) = \frac{1}{1+exp(-w^\top x)}
$$

(w为增广权值向量，x为增广特征向量，包含偏置)

则标签$y=0$的后验概率为：

$$
p(y=0|x) = 1 - p(y=1|x) = \frac{\exp(-w^\top x)}{1+\exp(-w^\top x)}
$$

结合上述两个公式，我们可以发现：

$$
w^\top x = \log\frac{p(y=1|x)}{1-p(y=1|1)} = \log\frac{p(y=1|x)}{p(x=0|x)}
$$

可以发现$f(x)$的值等于样本正反例后验概率比值的对数，也就是对数几率。所以Logistic回归可以看作预测值为标签的对数几率的回归模型。

参数学习方法

Logistic回归解决分类问题，使用交叉熵作为损失函数，使用梯度下降更新参数。

对于给定的N个训练样本$\{x^{(n)},y^{(n)}\}_{n=1}^N$，用$Logistic$回归模型对每个样本进行预测，输出其标签为1的后验概率，记作$\hat{y}^{(n)}$：

由于$y^n \in \{0,1\}$，样本$(x^{(n)},y^{(n)})$的真实条件概率可以表示为：

$$
P_r(y^{(n)}=1|x^{(n)}) = y^{(n)}
$$
$$
P_r(y^{(n)}=0|x^{(n)}) = 1 - y^{(n)}
$$

构造损失函数(交叉熵)：

$$
\mathcal{R}(w) = -\frac{1}{N}\sum_{n=1}^{N}\big((p_r(y^{(n)}=1|x)\log\hat{y}^{(n)} + p_r(y^{(n)}=0|x)\log(1-\hat{y}^{(n)})\big)
$$

应用经验风险最小化原则，$\mathcal{R}(w)$关于参数w的偏导数为：

$$
\begin{aligned}
\frac{\partial\mathcal{R}(w)}{\partial w} &= -\frac{1}{N}\sum_{n=1}^{N}\big(y^{(n)}\frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{\hat{y}^{(n)}}x^{(n)} - (1-y^{(n)})\frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{1-\hat{y}^{(n)}}x^{(n)}\big) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\big(y^{(n)}(1-\hat{y}^{n})x^{(n)} - (1-y^{(n)})\hat{y}^{(n)}x^{(n)}\big) \\
&== \frac{1}{N}\sum_{n=1}^{N}x^{(n)}(y^{(n)}-\hat{y}^{(n)})
\end{aligned}
$$

采用梯度下降法，$Logistic$回归的训练过程为：初始化$w_0 \leftarrow 0$，然后通过下式来迭代更新参数：

$$w_{t+1} \leftarrow w_t + \alpha \frac{1}{N} \sum_{n=1}^{N} x^{(n)} \left(y^{(n)} - \hat{y}_{w_t}^{(n)}\right)$$

其中α是学习率，$\hat{y}_{w_t}^{(n)}$是当参数为w_t时，Logistic回归模型的输出。

3.Softmax回归

Softmax回归可以看作多分类的Logistic回归。

Softmax函数：

$$Softmax(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{N} \exp(x_j)}$$

对x_i的偏导数为：

$$\begin{aligned}
\frac{\partial Softmax(x_i)}{\partial x_i} &= \frac{\partial \exp(x_i) \times [\sum_{j=1}^{N} \exp(x_j)]^{-1}}{\partial x_i} \\
&= \frac{\partial \exp(x_i) \times [\sum_{j=1}^{i-1} \exp(x_j) + \exp(x_i) + \sum_{j=i+1}^{N} \exp(x_j)]^{-1}}{\partial x_i} \\
&= \exp(x_i) \times [\sum_{j=1}^{N} \exp(x_j)]^{-1} + (-1) \times \exp(x_i) \times [\sum_{j=1}^{N} \exp(x_j)]^{-2} \times \exp(x_i) \\
&= \exp(x_i) \times [\sum_{j=1}^{N} \exp(x_j)]^{-1} - \exp(x_i)^2 \times [\sum_{j=1}^{N} \exp(x_j)]^{-2} \\
&= Softmax(x_i) - Softmax(x)^2 \\
&= Softmax(x_i) \times (1 - Softmax(x_i))
\end{aligned}$$

对于多分类问题$y \in 1, 2, \ldots, C$可以有C个取值，给定一个样本x，Softmax回归预测的属于类别c的条件概率为：

$$p(y = c|x) = \frac{\exp(w_c^\top x)}{\sum_{c'=1}^{C} \exp(w_{c'}^\top x)}$$

在Softmax回归中，模型的输出为一个C维的向量，分别表示对属于每个类别的概率的预测值。因此决策函数可以写作：

$$\hat{y} = \arg\max_{c=1}^{C} p(y = c|x) = \arg\max_{c=1}^{C} w_c^\top x$$

参数学习方法

Softmax回归同样使用交叉熵作为损失函数，用梯度下降来优化参数。

用C维`one-hot`向量$y \in \{0, 1\}^C$来表示类别标签，对于类别c，其类别标签向量为：

$$y = [I(1 = c), I(2 = c), \ldots, I(C = c)]^\top$$

根据定义构造风险函数：

$$\begin{aligned}
\mathcal{R}(w) &= -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \log \hat{y}_c^{(n)} \\
&= -\frac{1}{N} \sum_{n=1}^{N} (y^{(n)})^\top \log \hat{y}^{(n)}
\end{aligned}$$

风险函数$\mathcal{R}(w)$关于w的梯度：

$$\mathcal{R}(w) = -\frac{1}{N}\sum_{n=1}^{N} x^{(n)}(y^{(n)} - \hat{y}^{(n)})^{\top}$$

求解过程：

根据上文Softmax导数的结果，将其改写为向量式：

$$\frac{\partial Softmax(x_i)}{\partial x_i} = \mathrm{diag}(y) - yy^{\top}$$

若上式$x_i = w^{\top}x = [w_1^{\top}x, w_2^{\top}x, \ldots, w_C^{\top}x]^{\top}$，则$\frac{\partial w^{\top}x}{\partial w_c}$为第$c$列为$x$，其余为0的矩阵，即：

$$\begin{aligned}\frac{\partial w^{\top}x}{\partial w_c} &= [\frac{\partial w_1^{\top}x}{\partial w_c}, \frac{\partial w_2^{\top}x}{\partial w_c}, \ldots, \frac{\partial w_C^{\top}x}{\partial w_c}]^{\top} \\ &= [0, 0, \ldots, x, \ldots, 0]\end{aligned}$$

令$z = w^{\top}x$，那么根据链式求导法则：$\mathcal{L}^{(n)}(w) = -(y^{(n)})^{\top}\log\hat{y}^{(n)}$关于$w_c$的导数为：

$$\begin{aligned}\frac{\partial \mathcal{L}^{(n)}(w)}{\partial w_c} &= -\frac{\partial\big((y^{(n)})^{\top}\log\hat{y}^{(n)}\big)}{\partial w_c} \\ &= -\frac{\partial z^{(n)}}{\partial w_c}\frac{\partial \hat{y}^{(n)}}{\partial z^{(n)}}\frac{\partial \log\hat{y}^{(n)}}{\partial \hat{y}^{(n)}}y^{(n)} \\ &= -\mathbb{M}_c(x^{(n)})(\mathrm{diag}(\hat{y}^{(n)}) - \hat{y}^{(n)}(\hat{y}^{(n)})^{\top})(\mathrm{diag}(\hat{y}^{(n)}))^{-1}y^{(n)} \\ &= -\mathbb{M}_c(x^{(n)})(I - \hat{y}^{(n)}1_C^{\top})y^{(n)} \\ &= -\mathbb{M}_c(x^{(n)})(y^{(n)} - \hat{y}^{(n)}1_C^{\top}y^{(n)}) \\ &= -\mathbb{M}_c(x^{(n)})(y^{(n)} - \hat{y}^{(n)}) \\ &= -x^{(n)}[y^{(n)} - \hat{y}^{(n)}]_c\end{aligned}$$

故：

$$\frac{\partial \mathcal{L}^{(n)}(w)}{\partial w} = -x^{(n)}(y^{(n)} - \hat{y}^{(n)})^{\top}$$

采用梯度下降法，则训练过程为：初始化$w_0 \leftarrow 0$，迭代更新：

$$w_{t+1} \leftarrow w_t + \alpha\Big(\frac{1}{N}\sum_{n=1}^{N} x^{(n)}(y^{(n)} - \hat{y}_{w_t}^{(n)})^{\top}\Big)$$

α为学习率。

4.感知机

感知机是一种基于错误驱动在线学习的简单二分类线性模型。

$$\hat{y} = \mathrm{sgn}(w^{\top}x)$$

给定N个样本的训练集：$\{x^{(n)}, y^{(n)}\}_{n=1}^{N}$，其中$y^{(n)} \in \{+1, -1\}$，感知机尝试找到一组参数$w^*$，使得对于每个样本$(x^{(n)}, y^{(n)})$有：

$$y^{(n)}w^{*\top}x^{(n)} > 0, \forall n \in \{1, \ldots, N\}$$

参数学习方法

感知机的参数学习方法是直接定义的：初始化权重向量$w \leftarrow 0$，每分错一个样本(x, y)时，就用这个样本来更新权重：

$$w \leftarrow w + yx$$

根据以上定义反推感知机的损失函数：

$$\mathcal{L}(w; x, y) = max(0, -yw^\top x)$$

采用随机梯度下降更新参数，每次更新的梯度为：

$$\frac{\partial \mathcal{L}(w; x, y)}{\partial w} = \begin{cases} 0 & \text{if } yw^\top x > 0 \\ -yx & \text{if } yw^\top x < 0 \end{cases}$$

5.支持向量机

支持向量机(Support Vector Machine, SVM)是一个经典的二分类算法，其找到的分割超平面具有更好的鲁棒性，因此广泛应用在很多任务上，并表现出很强优势。

给定一个二分类器数据集$\mathcal{D} = \{(x^{(n)}, y^{(n)}\}_{n=1}^N$，其中$y_n \in \{+1, -1\}$，如果两类样本是线性可分的，即存在一个超平面：

$$w^\top x + b = 0$$

将两类样本分开，那么对于每个样本都有$y^{(n)}(w^\top x + b) > 0$。

数据集$\mathcal{D}$中每个样本$x^{(n)}$到分割超平面的距离为：

$$\gamma^{(n)} = \frac{|w^\top x^{(n)} + b|}{||w||} = \frac{y^{(n)}(w^\top x^{(n)} + b)}{||w||}$$

我们定义间隔γ为整个数据集$\mathcal{D}$中所有样本到分割超平面的最短距离：

$$\gamma = \min_n \gamma^{(n)}$$

如果间隔γ越大，其分割超平面对两个数据集的划分越稳定，不容易受到噪声等因素的干扰。支持向量机的目标是寻找一个超平面(w^*, b^*)使得γ最大，即下列约束问题：

$$\begin{aligned} \max_{w,b} \quad & \gamma \\ \text{s.t.} \quad & \frac{y^{(n)}(w^\top x^{(n)} + b)}{||w||} \geq \gamma, \forall n \in \{1, \ldots, N\} \end{aligned}$$

由于同时对w, b缩放不会改变样本$x^{(n)}$到分割超平面的距离，我们可以限制$||w|| \cdot \gamma = 1$，则公式等价于：

$$\begin{aligned} \max_{w,b} \quad & \gamma \\ \text{s.t.} \quad & y^{(n)}(w^\top x^{(n)} + b) \geq 1, \forall n \in \{1, \ldots, N\} \end{aligned}$$

数据集中所有满足$y^{(n)}(w^\top x^{(n)} + b) = 1$的样本点，都称为支持向量。

参数学习方法

将支持向量积的公式改写为凸优化形式：

$$\begin{aligned}&\min_{w,b}\ \ \frac{1}{2}||w||^2\\&\text{s.t. }\ 1-y^{(n)}(w^\top x^{(n)}+b)\leq 0,\forall n\in\{1,\ldots,N\}\end{aligned}$$

使用拉格朗日乘数法，构造拉格朗日函数：

$$\Lambda(w,b,\lambda)=\frac{1}{2}||w||^2+\sum_{n=1}^{N}\lambda_n\big(1-y^{(n)}(w^\top x^{(n)}+b)\big)$$

计算$\Lambda(w,b,\lambda)$关于w,b的导数：

$$\begin{aligned}\frac{\partial\Lambda(w,b,\lambda)}{\partial w}&=\frac{\partial\big[\frac{1}{2}||w||^2+\sum_{n=1}^{N}\lambda_n\big(1-y^{(n)}(w^\top x^{(n)}+b)\big)\big]}{\partial w}\\&=w-\sum_{n=1}^{N}\lambda_n y^{(n)}x^{(n)}\\\frac{\partial\Lambda(w,b,\lambda)}{\partial b}&=\frac{\partial\big[\frac{1}{2}||w||^2+\sum_{n=1}^{N}\lambda_n\big(1-y^{(n)}(w^\top x^{(n)}+b)\big)\big]}{\partial b}\\&=\sum_{n=1}^{N}\lambda_n y^{(n)}\end{aligned}$$

令$\Lambda(w,b,\lambda)$关于w,b的导数等于0，可得：

$$\begin{aligned}w&=\sum_{n=1}^{N}\lambda_n y^{(n)}x^{(n)}\\0&=\sum_{n=1}^{N}\lambda_n y^{(n)}\end{aligned}$$

结合拉格朗日函数及上式：原问题等价于：

$$\begin{aligned}&\min_{w,b}\ \ \frac{1}{2}||w||^2,w=\sum_{n=1}^{N}\lambda_n y^{(n)}x^{(n)}\\&\text{s.t. }\ 1-y^{(n)}(w^\top x^{(n)}+b)\leq 0,\ \sum_{n=1}^{N}\lambda_n y^{(n)}=0,\ \forall n\in\{1,\ldots,N\}\end{aligned}$$

构造拉格朗日对偶函数：

$$\begin{aligned}\Gamma(\lambda)&=\frac{1}{2}||w||^2+\sum_{n=1}^{N}\lambda_n\times[1-y^{(n)}(w^\top x^{(n)}+b)]\\&=\frac{1}{2}w^\top w-\sum_{n=1}^{N}\lambda_n y^{(n)}w^\top x^{(n)}-\sum_{n=1}^{N}\lambda_n y^{(n)}b+\sum_{n=1}^{N}\lambda_n\\&=\frac{1}{2}w^\top\sum_{n=1}^{N}\lambda_n y^{(n)}x^{(n)}-w^\top\sum_{n=1}^{N}\lambda_n y^{(n)}x^{(n)}+\sum_{n=1}^{N}\lambda_n\\&=-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_m\lambda_n y^{(m)}y^{(n)}(x^{(m)})^\top x^{(n)}+\sum_{n=1}^{N}\lambda_n\end{aligned}$$

根据KKT条件中的互补松弛条件，最优解满足：

$$\lambda_n^*(1-y^{(n)}(w^{*\top}x^{(n)}+b^*))=0$$

如果样本$x^{(n)}$不在约束边界上$\lambda_n^*=0$，约束失效；如果在约束边界上，样本点即支持向量，即距离决策平面最近的点。

只要得到λ^*即可通过得到w^*, b^*，则最优参数的支持向量机决策函数为：

$$\begin{aligned} f(x) &= \operatorname{sgn}(w^{*\top}x+b^*) \\ &= \operatorname{sgn}\big(\sum_{n=1}^{N}\lambda_n^* y^{(n)}x^{(n)}+b^*\big) \end{aligned}$$