三、线性模型

2016年2月13日 星期六 下午3:18

一、简介

二、线性回归

首先，对应的任务是回归任务。

f(x) = wx + b 的形式。

损失函数：均方误差，这个损失函数是回归问题中常用的损失函数。

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

他对应着欧氏距离，希望估计值与实际值的欧氏距离最小化，通过均方误差最小化求解模型的方法称为最小二乘法。

优化方法：最小化均方损失（最小二乘法），对损失求w和b导的，反馈更新。这里优化的函数是凸函数。

闭式解（closed-form）

由于线性回归的优化过程是凸函数，所有一定存在最优解。而且其实我们可以直接推出来，这个解我们称作闭式解或解析解。

1）如果不考虑x、w是一个多维向量的形式（而仅仅是一个数，这种最基本的形式）。做法就是对损失函数正常求导，令导数=0，解出w和b。线性回归的解析解具体结果。见机器学习-周志华P54

2）通常使用方式，x、w是一个多维向量，这时候，整个训练集可以表示为一个矩阵，m*d，m是训练集个数，d是特征维度。求解原理不变，也是求导，只不过这里用矩阵变换。过程见机器学习-周志华P55 或 CS299_1 pdf

闭式解结果

$$\theta = (X^TX)^{-1}X^T\vec{y}.$$

$$h(x) = \sum^{n}\theta_i x_i = \theta^T x$$

$$\sum_{i=0}$$

三、广义线性模型

广义线性模型，Generalized Linear Models (GLMs) 。

线性模型的形式虽然简单，但是有多种变换，比如y = wx+b是将x做线性变换拟合y，但是对于其他尺度的预测值呢，比如y是指数尺度上的。我们可以使用y = exp（ wx + b ），这也是线性模型体系。同样的，对于y = ln（ wx+b ）等等都是线性模型。

更一般化的，对于单调可微的函数g（ x ），令
y = g-1（ wx + b ）即 g（ y ）= wx + b，这样得等到模型称为广义线性模型。g（ x ）称为联系函数。逻辑回归、线性回归均是广义线性模型的特例。

（广义线性模型部分，机器学习-周志华简单提了一下，CS229_1 pdf中有详细介绍）
待补充CS229_1 pdf中更详细的！！！

四、逻辑回归

逻辑回归（logistic regression、逻辑斯谛克回归），机器学习-周志华中翻译为“对数几率回归”。
虽然名字叫“回归”，但是他是个分类模型。

公式：

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

对比：
他是广义线性模型的一个特例，g-1（ x ）为sigmoid

他是广义线性模型的一个特例，g^{-1}（x）为sigmoid。

Q 为什么使用sigmoid？

A 能将输出值做成0~1直接概率形式，方便求导求解没事任意阶可导的凸函数，许多数值优化算法（如，sgd、Newton Method等）都能求出最优解。

相比贝叶斯分类器，他不需要事先假设数据分布。

优化过程：

使用“极大似然法”，对“训练集所有样本上，预测正确的概率”最大化。

$$
\begin{aligned}
L(\theta) &= p(\vec{y} \mid X; \theta) \\
&= \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{m} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{1-y^{(i)}}
\end{aligned}
$$

在这里为了方便，转化为对数似然。

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))
\end{aligned}
$$

Q 求解过程，为什么不用最小二乘法直接求导？

A 最小二乘法的loss funtion适合均方误差，

至此，l(β)是待求的式子，这个式子是关于β的高阶连续可导凸函数，也是有解析解（闭式解）的。已经可以根据经典的数值优化算法求解，如牛顿法、sgd（进一步公式见机器学习-周志华P59）

另外，逻辑回归还有多分类的形式，多项逻辑回归（multi-mominal logistic regression），详见统计学习方法P79

五、感知器

感知器，感知机，perceptron。

注意，感知器属于“广义线性模型”族的模型。因为不满足广义线性模型要求的g（x）连续可微。

主要求的g（x）连续可微。

判别式：$y = \mathrm{sign}(wx + b)$

感知器的这个分类形式，可以看做是神经网络中的一个神经元。

优化算法：

回顾逻辑回归和线性回归的优化算法。

逻辑回归：对全体训练集上概率做极大似然，得到最大化的目标L，对L求偏导，得到解。

线性回归：根据均方误差本身特点，最小化均方误差，相当于最小化预测值和真实值的欧氏距离，得到最大化目标L，对L求导得到解。

感知机优化的时候最自然的想法是极大似然或者最小化误分类个数，但是由于sign函数对于w和b不是连续可导的，行不通。所以，我们采用最小化“误分类点到超平面的总距离”

总距离 $L = -\sum y_i（wx_i + b）$

求导可以很容易得到梯度公式，用sgd进行学习。

L 对于w和b是连续可导的，但是不是凸函数。所以没有解析解（闭式解）。

但是统计学习方法给出了算法收敛性的证明，可以得到以下结论：

A 对于线性可分的数据，经过有限次数的迭代，可以得到将数据完全正确分开的超平面，迭代是收敛的

B 感知器存在很多个解，这些解即依赖于初值，也依赖于训练顺序

C 如果想得到唯一解，需要增加限制条件（这就是SVM）

D 对于线性不可分的数据，不收敛

（感知器部分详见统计学习方法）

七、最大熵模型

（详见三、7 最大熵模型）

六、线性判别分析

线性判别分析不是线性模型，只不过放在这一部分讲述。

（详见三、6线性判别分析）

八、多分类

方法有两种：1）直接使用多分类器 2）通过多个二分类器组合，达到多分类的效果

机器学习-周志华中重点介绍了2），包括OvO（One vs One）、OvR（One vs Rest）、MvM（Many vs Many）

OvO（One vs One），使用k（k-1）/2个分类器，k是类数。任意两个类i，j之间都有一个分类器。所有的分类器投票，得票最多的类是最终的预测结果。

OvR（One vs Rest），使用k个分类器，k是类数。每个分类器负责判断“这个样本是i类的/不是i类的”。如果多个分类器之间冲突，比较分类器输出的概率值/置信度。

MvM（Many vs Many），每个分类器将若干个类视为“正”，若干个类视为“负”。每个类对应着一个编码，这个编码上记录着在各个分类器是属于“正“还是“负”。预测的时候，使用分类器预测出样本的编码，和各个类的编码比较，编码最相似的类作为最终的预测结果。这个相似的程度可以用海明距离等衡量。通常编码中还会有纠错码等。

九、分类不均衡问题

解决方案：

1、再缩放

由于预测值y/（1 - y）的比例应该正比于训练集正负例的比例，在假设“训练集无偏”的情况下，训练集正负例比例就是真实的正负例比例。

即 y/（1-y）> 正例个数/负例个数，则预测为正例；反之，负例。

我们为了预测结果中找到还原真实正负例比例，预测时做些改动，把y/（1-y）乘上负例个数/正例个数，还原实际比例。

不实用！因为实际问题中，常常“训练集是无偏的”假设不成立，也就是说乘以的系数有问题。

2、对训练集过多的样本“欠采样”

3、对训练集过少的样本“过采样”

4、阈值移动

5、另外，代价敏感学习，也就是让不同的类别判错的代价不一样，一般是修改loss function

十、其他

1、线性模型判别函数和判别面（模式分类）

对于g（x）=wx+b，如果是二分类问题，g（x）=0可以看做是判定面/分类面，称作“超平面”。w是超平面的发向量，和超平面上任意向量都是正交的。g（x）可以看做是特征空间中样本点x到超平面的距离的一种量度。