机器学习中的常见概念

在这里给大家罗列一下监督学习中常见的概念.

偏置量方差与准确度和泛化能力的关系

机器学习中一般偏置量bias指的是预测值与真实值之间的差的均值,而方差variance则是指的预测值在不同样本上的表现差异的大小. 因此偏置量可以表现模型的准确度,偏置量越小说明准确度越高;而方差则可以表现模型的泛化能力,方差越小说明泛化能力越强.

残差

残差是估计值与真实值之间的差

带标签的d维数据

数据有d个预测变量和标签,$\{\vec{x}, y : \vec{x} \in R^d, y \in Y\}$,其中$Y$为标签集

二元和多元

如果标签集有两个元素,叫做二元;如果含有更多元素,叫做多元

分类器

有监督学习的分类问题中,面对带标签的数据,我们学习到的用来预测标签的模型叫做分类器

生成式分类器和判别式分类器

生成式分类器和判别式分类器间最大的区别是他们可以被用来回答的问题.一个分类器可以用于回答如下6种问题:

1. 一个给定的输入最可能的标签是什么?
2. 对于一个给定的输入,给定一个标签,输入对应给定这个标签的可能性有多大?
3. 最有可能的输入值是什么?
4. 一个给定的输入值可能性有多大?
5. 某个特定的输入有某个特定标签的可能性有多大?
6. 对于一个可能有两个值中的一个值的输入,最可能的标签是什么?

生成式分类器可以回答上面全部的问题,而判别式分类器就只能回答前2个问题.

从这个角度讲生成式要比判别式强大,但强大是有代价的,生成式模型有更多的自由参数需要学习,而训练集的大小是固定的,因此相比较而言生成式分类器更难找到最佳的参数.

常见的生成式分类器有:

- 朴素贝叶斯分类器

常见的判别式分类器有:

- 逻辑回归分类器
- 最大熵分类器
- SVM分类器

超平面

d维线性空间$S = \{\vec{x} = (x_1, x_2, x_3, \ldots, x_d), x_i \in -\infty, \infty \forall i \in [1, d]\}$中的$d-1$维子空间,比如$x_1 * w_1 + x_2 * w_2 + \ldots + x_d * w_d = -b$,通常$b = 0$这个线性方程确定的平面就是d维空间中的一个超平面.注意,如果$b \neq 0$,我们可以将d维数据增加一维,使得$x_(d+1) = 1, w_(d+1) = b$.

线性可分

d维线性空间中,一组二元的d维数据集,如果可以根据标签被一个超平面完美得分割,我们就说这个数据集线性可分

分隔超平面

一组带标签的d维数据线性可分,那个超平面就叫做分隔超平面

线性分类器

分类问题中通过预测变量的线性组合来做出分类决策的模型

学习理论

人工智能的一个分支,用来研究机器学习算法的设计和分析

错误边界

在学习理论中,错误边界是一个机器学习算法收敛需要的更新数或者收敛前犯错数的上界

函数距离和几何距离

(点到超平面的)

在一个d维空间S中,一个超平面H由$x_1 * w_1 + x_2 * w_2 + \ldots + x_d * w_d = 0$决定,点$\vec{x} = (a_1, a_2, a_3, \ldots, a_d)$到H的函数距离为$||a_1 * w_1 + a_2 * w_2 + \ldots + a_d * w_d||$,几何距离为$\frac{||a_1 * w_1 + a_2 * w_2 + \ldots + a_d * w_d||}{||(w_1, x_2, \ldots, x_d)||}$

凸子集

拓扑和几何上,如果一个集合S满足,$\forall x_1, x_2 \in S$,对于$\forall 0 < t < 1, t * x_1 + (1 - t) * x_2 \in S$,通俗地讲,如果集合内任何两点连成的直线都在集合内,则集合S为凸.

凸函数

定义在某个线性空间的凸子集S上的函数f,如果满足
$\forall x_1, x_2 \in S, \forall 0 < t < 1, f(t * x_1 + (1 - t) * x_2) <= t * f(x_1) + (1 - t) * f(x_2 f)$,通俗地讲,如果函数图像任意两点之间的直线都在函数图像之上,则f为凸函数.

优化

优化是数学的一个分支,求解一个函数的极值,有时会有等式或者不等式的约束条件.

$min f(x)$

subject to $g_i(x) \geq 0, i \in [1, m]$

$h_j(x) = 0, j \in [1, l]$

其中$f(x)$叫做目标函数,$g_i(x), h_j(x)$分别为不等式和等式约束.当然,我们可以把等式约束用两个不等式约束$h_j(x) \geq 0, -h_j(x) \geq 0$代替.

凸优化

优化中,当目标函数和不等式约束函数都是凸的,该优化就是凸优化.如果我们将等式约束单独列出,那么要求等式约束是仿射的.因为$h_j(x) \geq 0, -h_j(x) \geq 0$,可以得出 $h_j(x)$既凸又凹,所以是线性的.

拉格朗日乘子法

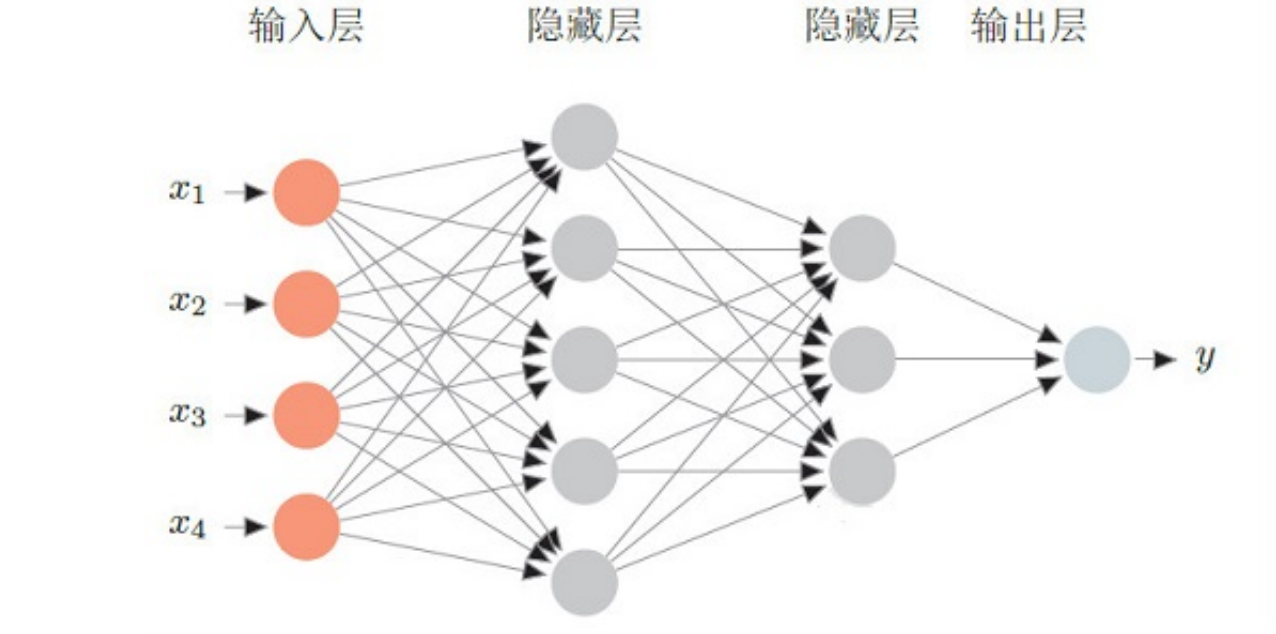
只带等式约束的优化中,基于隐函数定理,通过给等式约束添加拉格朗日乘子,对各个变量和拉格朗日乘子求一阶导数,来求得极值的方法.

KKT条件

目标函数非线性有不等式约束条件下的优化中,存在最优解的一阶充要条件:目标函数和不等式约束函数为凸,等式约束仿射.

前馈神经网络和循环神经网络

如图1,神经网络中神经元如果没有形成环,这个神经网络就属于前馈神经网络,反之则属于循环神经网络



损失函数

监督式机器学习算法是利用带标签的数据集训练出一个模型,来获得一种关于分类或者回归的知识,从而推广到新的数据.这里存在一个悖论,bias-variance dilemma (https://en.wikipedia.org/wiki/Bias–variance_tradeoff),也就是监督式学习中不可能同时降低两种错误:第一种错误bias来源于欠拟合,也就是模型偏离了数据集;第二种错误variance是因为过拟合,也就是模型过度考虑了用于训练的数据集而失去了推广能力.

通俗地讲,如果一个模型太复杂,虽然它准确得拟合了手上的数据集,但它的推广能力(泛化能力)就要打折扣.如果一个模型太简单,也许可以推广到新的数据(新数据是未知的),但对手上数据集的解释力很差.

因为新数据是未知的,所以训练模型最常犯的错误是过拟合,为了降低过拟合的风险,我们有5种对策:

1. 正则化,最典型的做法是损失函数添加正则化项,模型复杂度越高,正则化项越大.
2. 训练集拆分.也就是将训练集分为训练集,测试集.新的训练集一部分用来训练模型,测试集用来当做新数据,检验模型推广能力.
3. 交叉训练,也就是多次随机得将原训练集拆分成新的训练集和验证集,在新的训练集上训练模型,通过模型在验证集上的表现来选取超参.
4. 使用数据增强技术人为的增大数据集.
5. 在神经网络中使用dropout在反向计算时随机屏蔽一定比例的权重变化.

这两类错误在模型训练时体现在损失函数$L(x, y, \theta)$上,损失函数往往由两部分构成,经验风险R_{emp}和正则化项(模型复杂度的惩罚项)$r(d)$.

其中经验风险对应的是模型在训练集上的错误,比如

1. 逻辑回归中的logit loss或log odds function,$R_{emp} = \sum_{i=1}^{N}(y_i * (w * x_i + b) - log(1 + e^{w*x_i+b}))$.其中N为样本大小,$y_i$和$\hat{y}_i$分别是第i个样例的真实值和预测值.最小化logit loss可以得到与样本集的真实概率分布在KL divergence意义上最接近的概率分布..recall that逻辑回归中预测的

$P(y_i = 1|x_i) = \frac{1}{1+e^{-(w*x_i+b)}}$,Logistic损失实际上就是二分类下的cross-entropy (https://en.wikipedia.org/wiki/Cross_entropy)

2. hard margin SVM中的$R_{emp} = |w|^2$,既是经验风险也是正则化项,是基于函数距离和几何距离的关系,用于最小化几何边界,可以让模型的错误边界最小.详情参看SVM的章节 (https://github.com/HAOzj/Classic-ML-Methods-Algo/blob/master/ipynbs/supervised/SVM.ipynb).
3. soft margin SVM中的hinge loss function $R_{emp} = \sum_{i=1}^{N} max(0, 1 - \hat{y_i} * y_i)$,可以最小化slack variables,也就是让误分类更少.它同样也扮演了正则化项的角色,因为是L1正则,所以得到的解相比平方损失这种L2正则也更加稀疏
4. 线性回归中的平方损失$R_{emp} = \sum_{i=1}^{N} (\hat{y_i} - y_i)^2$,这样是可微凹函数,可以得到的均值是无偏的,但对离群点(outliers)敏感.
5. 绝对损失$R_{emp} = \sum_{i=1}^{N} \hat{y_i} - y_i$,这样对离群点不如平方损失敏感,得到的是中点(median)无偏的,但不可微,得到的解也比较稀疏.

而正则化项$r(d)$往往和参数的大小以及模型的复杂度有关,比如多项式模型的正则化项就比线性模型的大.

In [ ]: