machine learning summary

第一章

machine learning 的概念

acquiring skill with experience accumulated/computed from data

improving some performance measure with experience computed from data

分类

二元分类、多元分类，回归

无监督，半监督 全监督

1. 无监督学习

无监督学习是指训练数据没有标签或类别信息的机器学习方法。在无监督学习中，模型需要从数据中自动学习数据的结构和模式，以便将数据划分为不同的簇或进行降维等操作。常见的无监督学习算法包括聚类算法和降维算法。

2. 半监督学习

半监督学习是指训练数据仅有一部分数据有标签或类别信息的机器学习方法。在半监督学习中，模型需要使用标记数据来指导学习过程，以便更好地理解未标记数据的结构和模式。常见的半监督学习算法包括半监督聚类和标签传播算法等。

3. 全监督学习

全监督学习是指训练数据有标签或类别信息的机器学习方法。在全监督学习中，模型需要使用标签数据来训练模型，以便进行分类、回归等任务。常见的全监督学习算法包括线性回归、决策树、神经网络等。

强化学习

batch learning（批量学习） online learning（在线学习） active learning（主动学习）

1. Batch Learning（批量学习）：

批量学习是指从一次性的数据集中训练模型的方式。在批量学习中，训练集的所有数据都被用来训练模型，然后通过调整模型的参数来最小化训练误差。批量学习通常用于离线训练，其优点是可以在一次训练中获得全局最优解，但是需要耗费大量的计算资源和时间。

2. Online Learning（在线学习）：

在线学习是指通过逐步地增量学习数据来训练模型的方式。在在线学习中，模型会不断地根据新的数据进行更新，以适应不断变化的环境和数据。在线学习通常用于实时训练，其优点是可以快速响应数据的变化，并且节省计算资源和时间。但是在线学习也存在一些挑战，例如如何确定合适的学习率、如何处理噪声数据等。

3. Active Learning（主动学习）：

主动学习是指通过选择最有价值的数据点来训练模型的方式。在主动学习中，模型会根据已有的数据对数据进行分类，然后选择最不确定或者最有代表性的数据点来进行标注，以提高模型的泛化能力和准确性。主动学习通常用于数据稀缺或者标注成本高昂的情况下，其优点是可以在少量的数据中取得很好的效果，但是需要合理选择标注数据的策略。

特征种类

数值型特征，分类型特征，有序性特征，时间性特征，文本型特征，土星刑特征

第二章线性回归

线性回归:

线性回归的回归方程形式可以表示为:

$y = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n + \epsilon$

$H_\theta(x) = \theta^T x$

其中，y 是响应变量（或因变量），$x_1, x_2, \ldots, x_n$ 是预测变量（或自变量），$w_0, w_1, w_2, \ldots, w_n$ 是回归系数，ϵ 是误差项（或随机误差）。线性回归的目标是找到一组回归系数 $w_0, w_1, w_2, \ldots, w_n$，使得预测变量 $x_1, x_2, \ldots, x_n$ 与响应变量 y 之间的线性关系最为接近。

在实际应用中，线性回归模型可以通过最小化平方误差来拟合回归系数，也就是通过最小化以下损失函数来求解回归系数:

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

其中，θ 表示回归系数向量，m 表示样本数，$h_\theta(x^{(i)})$ 表示用回归系数向量 θ 对样本 $x^{(i)}$ 进行预测的结果，$y^{(i)}$ 是样本 $x^{(i)}$ 对应的真实值。最小化上述损失函数可以使用梯度下降等优化算法来求解最优的回归系数。

L2loss

L1范数误差:

$$\mathrm{MAE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\left|\mathrm{f}(\mathrm{x_i}) - \mathrm{y_i}\right|$$

优点:

- 无论对于什么样的输入值，都有着稳定的梯度，不会导致梯度爆炸问题，具有较为稳健性的解。
- 对于离群点不那么敏感。因为MAE计算的是误差y－f(x)的绝对值，对于任意大小的差值，其惩罚都是固定的。

缺点：

- MAE曲线连续，但是在y − f (x) = 0 y−f(x)=0y−f(x)=0处不可导。而且 MAE 大部分情况下梯度都是相等的，这意味着即使对于小的损失值，其梯度也是大的。这不利于函数的收敛和模型的学习。

L2范数误差（L2 loss)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$$

优点：MSE的函数曲线光滑、连续，处处可导，便于使用梯度下降算法，是一种常用的损失函数。 而且，随着误差的减小，梯度也在减小，这有利于收敛，即使使用固定的学习速率，也能较快的收敛到最小值。

缺点：当y yy和f (x) f(x)f(x)之间的差值大于1时，会放大误差；而当差值小于1时，则会缩小误差，这是平方运算决定的。MSE对于较大的误差（>1）给予较大的惩罚，较小的误差（<1）给予较小的惩罚。也就是说，对离群点比较敏感，受其影响较大。如果样本中存在离群点，MSE会给离群点更高的权重，这就会牺牲其他正常点数据的预测效果，最终降低整体的模型性能，甚至可能引发梯度爆炸

经验误差，泛化误差

通常使用均方误差（MSE）作为损失函数来衡量模型预测输出与真实输出之间的差距。

在机器学习中，通常将误差分为两种类型：经验误差和泛化误差。

1. 经验误差

经验误差是指模型在训练集上的误差，它是用训练数据集来评估模型的性能。在线性回归中，我们可以使用训练集来训练模型，并计算出模型在训练集上的MSE，这就是经验误差。

2. 泛化误差

泛化误差是指模型在未知数据上的误差，即在测试数据集上的误差。泛化误差是用来衡量模型在实际应用中的预测能力，因为我们通常关心的是模型在新数据上的性能。

实际上我们无法直接计算出模型的泛化误差，因为测试数据集通常是不可知的。因此，我们需要使用一些技巧来估计模型的泛化误差。其中最常用的技巧是交叉验证。通过将数据集分为训练集和验证集，并多次随机地交换它们的组合，从而获得模型在不同验证集上的误差估计值，进而对模型的泛化误差进行估计。

矩阵形式

加上L2后的解

在线性回归中加入 L2 正则化后，损失函数可以表示为：

$L(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$

其中，θ 表示回归系数向量，m 表示样本数，$h_\theta(x^{(i)})$ 表示用回归系数向量 θ 对样本 $x^{(i)}$ 进行预测的结果，$y^{(i)}$ 是样本 $x^{(i)}$ 对应的真实值，n 表示回归系数的个数（不包括偏置项 w_0），λ 是一个控制正则化强度的超参数。

在加入了 L2 正则化之后，优化的目标是最小化新的损失函数。为了避免过拟合，正则化项 $\frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$ 会对原来的损失函数进行惩罚，使得回归系数的值不能太大，从而限制模型的复杂度。

求解加入 L2 正则化后的线性回归问题可以使用梯度下降等优化算法来最小化新的损失函数，具体的求解方法可以参考岭回归（Ridge Regression）。

逻辑回归

sigmod函数

逻辑回归

条件概率

最大似然

求对数

求导

牛顿法求解

二分类

迭代优化两种方法：

- 确定性优化——给定问题的数据是准确已知的。

  优点：收敛速度快，收敛性号，通常可以找到全局最优解。

常见方法包括：梯度下降，牛顿法

- 随机优化——指在存在随机性的情况下，最小化或最大化目标函数的一系列方法。

  优点：可以处理大规模数据和复杂模型，可以避免陷入局部最优解，但是收敛速度慢，可能会出现收敛到次优解的情况

  常见的方法有：随机梯度嘉奖，随机牛顿法，遗传算法

梯度下降的一阶和二阶：

- 一阶方法:只使用梯度的方法。
- 二阶方法:也使用黑森矩阵的方法。

一阶展开式：

假设当前参数为 θ，目标函数为 $J(\theta)$，梯度为 $\nabla J(\theta)$，学习率为 α，则梯度下降的一阶展开式为：

$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$

其中 t 表示迭代次数。

二阶展开式：

二阶展开式在一阶展开式的基础上，加入了目标函数的二阶导数信息。假设当前参数为 θ，目标函数为 $J(\theta)$，梯度为 $\nabla J(\theta)$，Hessian 矩阵为 $H(\theta)$，学习率为 α，则梯度下降的二阶展开式为：

$\theta_{t+1} = \theta_t - \alpha (H(\theta_t))^{-1} \nabla J(\theta_t)$

其中 t 表示迭代次数，$(H(\theta_t))^{-1}$ 表示 Hessian 矩阵的逆矩阵。注意，这里的 Hessian 矩阵是目标函数 $J(\theta)$ 对参数 θ 的二阶导数构成的矩阵。在实际应用中，计算 Hessian 矩阵的逆矩阵比较困难，因此二阶展开式的使用比较少。

梯度下降的最优解全局，局部，学习率，步长的影响

牛顿法更新规则

- Motivation: to minimize the local second-order Taylor approximation of f.

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \approx \min_{\boldsymbol{x}} f(\boldsymbol{x}_t) + \nabla f(\boldsymbol{x}_t)^T (\boldsymbol{x} - \boldsymbol{x}_t) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}_t)^T \nabla^2 f(\boldsymbol{x}_t)(\boldsymbol{x} - \boldsymbol{x}_t)$$

- Take the derivative of $\boldsymbol{x}$ on both side, we have,

$$\frac{df(\boldsymbol{x})}{d\boldsymbol{x}} = \nabla f(x_t) + \nabla^2 f(\boldsymbol{x}_t)(\boldsymbol{x} - \boldsymbol{x}_t) = \boldsymbol{0}$$

- Update rule: suppose $\nabla^2 f(x_t)$ is positive definite,

$$\boldsymbol{x} = \boldsymbol{x}_t - [\nabla^2 f(x_t)]^{-1} \nabla f(x_t)$$

- 优点：

  更精确的局部近似值，

  收敛速度快得多。

- 缺点：

  需要计算二阶导数

  需要计算黑森逆(时间/存储消耗)

BGD，SGD，MBGD

BGD（batch-gradient-descent)（批量梯度下降）、

整个训练数据集被用来计算成本函数相对于模型参数的梯度，然后使用该梯度来更新模型参数

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

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

SGD (stochastic gradient-descent)（随机梯度下降）

每次仅使用一个训练样本来计算成本函数的梯度，并使用该梯度来更新模型参数。

$$\theta_j := \theta_j - \alpha (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

MBGD(mini-gradient-descent)（小批量梯度下降）

每次使用一个小批量的训练样本（通常为2-100个样本）来计算成本函数的梯度，并使用该梯度来更新模型参数。

1. BGD（批量梯度下降法）

   优点：

- 收敛性能稳定，每次更新的方向都是全局最优方向，能够保证达到全局最优解；
- 由于每次更新都是在整个数据集上进行的，因此能够获得较为精确的梯度信息。

缺点：

- 训练时间较长，因为需要处理整个数据集；
- 在数据量较大时，内存可能会受限，无法同时处理整个数据集；
- 无法处理动态数据，当新数据到来时需要重新训练模型。

2. SGD（随机梯度下降法）

   优点：

- 训练速度快，每次只需要处理一个样本或者一小批样本，迭代次数较少；
- 对于大规模数据集，SGD通常比BGD更加适用；
- 可以逐渐逼近全局最优解，在一定程度上避免了局部最优的问题。

缺点：

- 由于每次仅考虑一个样本或者一小批样本，因此难以准确估计梯度，容易受到噪声干扰；
- 随机性较强，会导致目标函数在更新过程中出现震荡或抖动的情况；
- 不能保证达到全局最优解，可能会陷入局部最优解。

3. MBGD（小批量梯度下降法）

   优点：

- 既可以处理大规模数据集，又比SGD更加稳定，能够更快地收敛；
- 能够平衡全局最优解和局部最优解的问题，具有较好的通用性；
- 实现起来比BGD和SGD都容易，可以利用矩阵运算进行优化。

缺点：

- 需要调整批量大小来平衡速度和准确度之间的权衡；
- 可能会陷入局部最优解。