MLPR第二讲学习笔记

by zijeff

机器学习技术分类

从机器学习的过程看，机器学习算法(或者机器学习模型)可分为以下几种：

- **有监督学习** (Supervised Learning)
- **无监督学习** (Unsupervised Learning)
- **半监督学习** (Semi-Supervised Learning)
- **强化学习** (Reinforcement Learning)

> **Tip**
>
> 监督学习、无监督学习和半监督学习的**区别**：
>
> - 监督学习处理的对象是所谓的有标签训练数据，它利用有标签的训练数据来学习一个模型，它的目标是用学到的模型给无标签的测试数据打上标签。**常见的分类和回归问题都属于监督学习。**
> - 无监督学习的训练数据**没有标签**，它自动从训练数据中学习知识，建立模型。
> - 半监督学习是监督学习和无监督学习相结合的一种学习方法，其**基本原则**是通过大量无标记数据辅助少量已标记数据进行学习，从而提高学习效果。

一些经典的有监督学习算法

- **K-近邻算法** (K-Nearest Neighbors, **KNN**)
- **线性回归** (Linear Regression)
- **逻辑回归** (Logistic Regression)
- **支持向量机** (Support Vector Machines, **SVM**)
- **决策树和随机森林** (Decision Trees & Random Forests)
- **神经网络** (Neural Networks)

一些经典的无监督学习算法

- **聚类算法：**
    1. **K-均值聚类**(K-Means)
    2. **层次聚类分析**(Hierarchical Cluster Analysis)
    3. **概率聚类分析**(Probabilistic Cluster Analysis)
- **降维算法：**
    1. **主成分分析**(PCA)
    2. **核主成分分析**(K-PCA)
    3. **局部线性嵌入**(LLE)
- **关联规则学习算法：**
    1. **Apriori**
    2. **Eclat**

> **Important**
>
> 由于这节课面向全校学生开设，所以我们将重点分析**K-近邻算法 (K-Nearest Neighbors, KNN)**和**线性回归 (Linear Regression)**这两种有监督学习算法。

K-近邻算法 (K-Nearest Neighbors, KNN)

- **KNN的定义：**所谓K-近邻法，就是给定一个训练数据集，对新的输入样本(样例/实例/模式)，在训练数据集中找到与该样本最邻近的K个样本，这K个样本的**多数**属于某个类，就把该输入样本分类到这个类中。当$K=1$时，**KNN**法便成了最近邻法，即寻找最近的那个邻居。
- **基本思想：**
  1. 产生训练集，使得训练集按照已有的分类标准划分成离散型数值类，或者是连续型数值类输出。
  2. 以训练集的分类为基础，对测试集每个样本寻找K个近邻，采用**欧几里得距离**作为样本间的**相似程度**的判决依据，相似度大的即为最近邻。一般近邻可以选择1个或者多个。
  3. 当类为连续型数值时，测试样本的最终输出为近邻的平均值；当类为离散型数值时，测试样本最终归为近邻类中个数最多的那一类。
- **K-近邻法的三个基本要素：**
  1. **K值的选择：**k值的选择会对算法的结果产生重大影响。k值较小意味着用较小的邻域中的训练样本进行预测，只有与输入样本较近的训练样本才会对预测结果起作用，但容易发生**过拟合 (over fitting)**；如果k值较大，优点是能减少"学习"的**估计误差 (estimation error)**，但缺点是"学习"的**近似误差 (approximation error)**会增大，这时与输入样本较远的训练样本也会对预测起作用，使预测发生错误。实际应用中，k值一般选择较小的数值。具体应用中，k值的选择通常需要通过大量的实验来确定。
  2. **距离度量方法：**距离的度量可采用**欧几里得距离**，也可采用更一般的L_p距离，即**闵可夫斯基距离（Minkowski distance）**。

     > **Note**
     >
     > 在第一讲笔记中，我们明确了L_p距离在$p=1$和$p=2$这两种情况下指的是**绝对值距离**和**欧几里得距离。**而当p趋向于正无穷时，也就是L_∞距离，即**切比雪夫距离。**这是一个非常直观的推断，当p趋向于正无穷大时，式子的值变化趋势将由其中的最大的项主导，所以我们可以认为**极限就是最大项的值。**详细的证明可以自行上网查证。

  3. **分类决策规则/回归标签值计算方法：**
     - 分类决策规则往往是**多数表决**，即待识别样本由K个最临近的训练样本中的多数类来决定待识别样本的类别。
     - 回归标签值计算常用**求均值函数**、**线性回归模型**和**局部加权线性回归模型。**
- **KNN算法描述：**
  1. 输入：训练用数据集T，以及等待识别的模式（样本）$\boldsymbol{x}$，我们需要做的便是把样本分到某一确定的类别中。值得注意的是，训练数据集表示为：

     $$T=\{(\boldsymbol{x_1},y_1),(\boldsymbol{x_2},y_2)\cdots(\boldsymbol{x_N},y_N)\},$$

     其中$\boldsymbol{x}_i,i\in\{1,2,\cdots,N\}$是具有若干特征的**样本向量**，$y_i\in\{\omega_1,\omega_2,\cdots,\omega_M\}$是该样本所属的**特定类别。**
  2. 输出：等待识别的模式（样本）$\boldsymbol{x}$所属的类别。

> **Important**
>
> 具体**求解方式**如下：

- 首先根据我们给定的**距离度量**，在训练集T中找出与$\boldsymbol{x}$最近邻的k个点，将涵盖这k个近邻点的**邻域**记为$N_k(\boldsymbol{x})$
- 然后在邻域$N_k(\boldsymbol{x})$中根据**分类决策规则**（常用多数表决规则），进而确定$\boldsymbol{x}$的类别y
- 抽象一下，我们可以得到计算$\boldsymbol{x}$的类别y的**一般数学公式**

$$y=\arg\max_{y_i}\sum_{\boldsymbol{x}_i\in N_k(x)}I(y_i=\omega_j),i\in\{1,2,\cdots,N\},j\in\{1,2,\cdots,M\}$$

上述式中的I为**指示函数**，当条件为真时取1，否则取0。

由于**KNN算法**所依赖的数学工具，导致该算法必然存在着如下**问题**。

> ⚠ **Warning**
>
> 由于**维度灾难(curse of dimensionality)**的原因，使得KNN算法易**过于拟合**。维度灾难是这样一种现象：对于样本数量大小稳定的训练数据集，随着其特征数量的增加，样本中有具体值的特征数量变量极其稀疏(大多数特征的取值为空)。直观地说，可以认为即使是最近的邻居，它们在高维空间的实际距离也是非常远的，因此**难以给出一个合适的类别标签判定**。

线性回归（Linear Regression）

还是那句话，由于这门课是面向全校学生开放的，所以我们接下来着重讨论**一元线性回归**。

> ⓘ **Note**
>
> (⊙﹏⊙)，🤔🤔🤔。既然着重要讲一元线性回归，可是一元线性回归的具体求解我们在高中已经系统学习过。所以这部分先空着吧，等笔者搞明白多元线性回归再回来补，绝对不是因为懒！