【python+机器学习0】用python实现机器学习介绍引言

我们已知机器学习是实现人工智能的一个重要手段，其中每个算法的学习能力是衡量智能行为的一个非常重要的因素。这种学习能力是指在遇到错误时进行自我校正、通过经验（历史数据）改善算法自身的性能并且通过深度学习自动获取和发现所需要的知识。

机器学习的最新阶段始于1986年，到目前已经得到广泛发展，可以在多领域交叉成熟应用。鉴于此，我们将借助python已有的五大数据分析的核心库：numpy、scipy、pandas、matplotlib、scikit-learn等，重点介绍机器学习中的相关算法实现和具体应用。

##### Python核心库的介绍与安装

|  |  |
| --- | --- |
| python库 | 基本介绍 |
| Numpy | Python的一个科学计算的基础库，可以存储和处理大型矩阵，重在数值计算，用于多维数组处理。 |
| Scipy | Python中科学计算程序的核心包，基于numpy，提供了一个在python中做科学计算的工具集。可以有效地计算numpy矩阵，让numpy和scipy协同工作。 |
| Pandas | Python中最具有统计意味的工具包，基于numpy的一种工具，为解决数据分析任务而创建，纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。 |
| Matplotlib | Python中最著名的绘图系统，很多其他的绘图例如seaborn（针对pandas绘图而来）也是由其封装而成。可以实现散点图，折线图，条形图，直方图，饼状图，箱形图等的绘制。 |
| scikit-learn | Python中使用Scikit-learn(简化为sklearn)这一库来处理机器学习，主要依赖于numpy, scipy和matplotlib库，其中机器学习模型十分丰富，需要根据问题的类型来选择适当的模型。 |

如果使用anaconda（强烈推荐）则已经集成了大量的数据分析库，只需要额外按照scikit-learn库即可。（Anaconda官网：https://www.continuum.io/downloads）

使用pip 命令安装

pip install -U scikit-learn

或conda 命令安装

conda install scikit-learn

否则的话需要手动依次安装，注意安装顺序为: numpy -> scipy -> matplotlib -> sklearn。

##### 机器学习算法的分类与介绍

在机器学习(Machine learning)中。主要有三类不同的学习方法：有监督学习(Supervised learning)、无监督学习(Unsupervised learning)以及半监督学习(Semi-supervised learning)，各自的介绍和主要的算法如下：

|  |  |
| --- | --- |
| 机器学习类型 | 主要的算法 |
| 有监督学习：分类和回归  训练集有输入和输出，相关算法可以从给定的训练集中学习出一个函数或模型，当新的数据到来时，可以根据这个函数预测对应结果。 | 最邻近分类算法、线性回归算法（普通最小二乘法、岭回归、逻辑回归等）、朴素贝叶斯算法、决策树、核支持向量机等 |
| 无监督学习：聚类和降维  训练集只有输入没有输出，相关算法实现目标是对数据中潜在的结构和分布建模，以便对数据作更进一步的学习。 | k-means聚类、主成分分析、因子分析、非负矩阵分解、流形学习等 |
| 半监督学习：介于前两者之间  训练集中有少量标注数据和大量未标注数据。相关算法设计的主要目的是实现具有良好性能的分类器。 | 聚类如果(cluster assumption)、流形如果(maniford assumption)等 |

##### Sklearn库中的标准数据集

使用python实现上述算法，我们将直接借助sklearn库中封装好的一些数据集。



在之后的分享中，我们将会一一介绍这些算法的基本理论、python库的具体使用以及相关注意事项。

监督学习：通过已有的一部分输入数据与输出数据之间的相应关系。生成一个函数，将输入映射到合适的输出，比如分类。

非监督学习：直接对输入数据集进行建模，比如聚类。

半监督学习：综合利用有类标的数据和没有类标的数据，来生成合适的分类函数。

