第5章

分析方法初步

5.1 机器学习基础

5.1.1 何为机器学习

1. 机器学习的定义

在日常生活中人们经常会根据经验来解决遇到的问题，例如看到朋友有黑眼圈，精神状态差，人们推测可能他没有休息好；看到天空颜色变暗，狂风骤起，燕子低飞，人们预测可能马上会下雨。为什么人们会做出这样的预测呢？因为人们已经总结了足够多的类似情况，所以在新的问题发生时，可以根据以往的经验做出较为准确的预测，例如人们根据经验学习到了燕子低飞这个特征与快要下雨相关联。

上面对于经验的利用以及新的结果的预测是通过人来实现的，那么计算机是否也可以模仿人来完成这个工作呢？机器学习就是一门致力于通过数据以及以往的经验，优化计算机程序性能的学科。机器学习是一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科，专门研究计算机怎样模拟或实现人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构使之不断改善自身的性能。

机器学习有 3 个要素，分别为**任务**（Task）、**性能**（Performance）、**经验**（Experience）。假设采用 T 代表任务，P 代表任务 T 的性能，E 代表经验，机器学习研究的主要内容是利用经验 E 通过“**学习算法**”（Learning Algorithm）提高任务 T 的性能 P，最终从经验中产生“**模型**”（Model）。下面举例说明机器学习的基本要素。

1）鸢尾花分类系统

（1）任务 T：对给定鸢尾花进行分类，即判断鸢尾花的品种。

（2）性能指标 P：分类的准确率。

（3）经验来源 E：大量的鸢尾花数据以及其对应的品种。

2）垃圾邮件分类系统

（1）任务 T：判断给定邮件是否垃圾邮件。

（2）性能指标 P：分类的准确度。

(3) 经验来源 E：大量的邮件数据以及其对应的类别。

2. 机器学习的应用

机器学习在很多应用领域具有十分广泛的应用，被证明拥有非常大的实用价值，尤其表现在以下几个方面[4]。

(1) **数据挖掘**：数据挖掘的主要任务是从大量的数据中通过算法搜索隐藏于其中信息的研究工作。数据挖掘目前主要应用于数据统计分析、销售数据、网络数据分析、流量数据分析、风险评估等多个方面。

(2) **计算机视觉**：计算机视觉是使用计算机及相关的设备对生物视觉的一种模拟，通过对采集的图片或者视频进行处理，最终获取所需被拍摄对象的数据和信息的一门学科。计算机视觉目前主要用于医学图像处理、导弹制导以及无人机和无人驾驶车辆等应用领域。

(3) **自然语言处理**：自然语言处理主要研究实现人与计算机之间通过自然语言进行有效通信的理论与方法。自然语言即人们日常使用的语言，通过自然语言处理，主要实现文本分类与聚类、信息检索和过滤、机器翻译等多种应用。

(4) **生物特征识别**：生物特征识别主要利用人体固有的生理特征(虹膜、指纹、声纹、DNA 等固有特征)或行为特征(步态、签名等习惯)实现个人身份鉴定的技术。目前较为火热的研究方向有人脸识别、亲子鉴定、指纹识别、虹膜识别等身份鉴定技术。

5.1.2　基本术语

机器学习是一类算法的总称，这些算法企图从大量历史数据中挖掘出其中隐含的规律，并用于对新的数据进行预测或者分类。更具体地说，机器学习可以看作是寻找一个函数，输入是大量的样本数据，输出是期望结果。

要进行机器学习，必须要获取经验 E，而计算机中经验一般以数据的形式存在，通过使用学习算法对数据进行学习最终可以获取模型。假定我们获取一批鸢尾花的数据，收集每个鸢尾花的花萼长度、花萼宽度、花瓣长度以及花瓣宽度 4 个值，单位为厘米。数据的组织形式例如(2,1,2,2),(3,1,2,2),…,(1,1,2,2)。这些数据的集合称为一个**数据集**(Data Set)，每条数据称为一个**样本**(Sample)或者**示例**(Instance)。其中花萼长度、花萼宽度等称为鸢尾花数据的**属性**(Attribute)或者**特征**(Feature)。属性上的值例如 2cm，被称为**属性值**。如果按照花萼长度、花萼宽度等 4 个属性建立四维空间，那么每一个鸢尾花都可以在空间中找到自己的坐标向量，所以示例也可以被称为**特征向量**(Feature Vector)。

从数据中学得模型的过程被称为**学习**或**训练**，在通过使用学习算法对数据进行训练的过程中，每一个样本被称为**训练样本**，训练样本组成的数据集被称为**训练集**。训练模型的目的是预测关于数据的某种潜在的规律，通过学习过程逐渐逼近这个规律。学习过程只依靠前面的样本的数据信息是不够的，需要知道拥有这些特征的数据到底会产生什么样的结果，例如当样本属性值为(2,1,2,2)时是山鸢尾，当样本属性值为(3,1,2,2)时是杂色鸢尾。其中样本信息的结果“山鸢尾”“杂色鸢尾”被称为**标记**(Label)。

获取对样本的属性值与样本的标记的组合之后，人们采用学习算法逐渐习得其内部存在的规律，实现学习任务。根据人们预测的结果不同可以将学习任务分为不同类别。如果需要预测的是离散值，例如“山鸢尾”“杂色鸢尾”“弗吉尼亚鸢尾”，则将此类任务称为**分类**(Classification)；如果预测的是连续值，例如鸢尾花的绽放程度为 0.9、0.1，则将此类任务称为**回归**(Regression)。学得模型后可以对未知的新样本进行预测，过程称为**测试**，被预测的样本称为**测试样本**。例如在习得 f 之后，对测试样本 x 进行测试，测试的标记为 $y=f(x)$。

除了对鸢尾花数据集进行回归与分类任务，还可以对鸢尾花数据集进行**聚类**(Clustering)，即将数据集分成若干个组，每个组称为一个**簇**(Cluster)。这些生成的簇可能对应一些潜在的概念划分，使得同一个类的样本相似，不同类的样本之间尽量不同。例如基于颜色的划分，“蓝色花”“紫色花”等，或者基于形状的划分，“花骨朵”“绽放的花”等。而究竟按照何种策略进行聚类是没有预先定义的，是其通过聚类算法在学习过程中自己生成的区分概念。

按照训练数据有无标记信息可以将学习任务大致分为两大类：**监督学习**(Supervised Learning)和**无监督学习**(Unsupervised Learning)，其中分类和回归是监督学习的代表，聚类是无监督学习的代表。

5.1.3 模型评估与性能度量

机器学习的目标是由训练集学得的模型可以适用于非训练集样本的预测。人们希望对于未出现在训练集中的数据采用预测模型也可以得到较好的预测效果。学得模型适用于新样本的能力称为**泛化**(Generalization)。模型设计虽然只是基于整个样本空间很少的部分进行预测，但是人们的目标是其在整个样本空间都有较好的预测能力，所以必须保证模型拥有较强的泛化能力。

机器学习的目标是使模型具有较好的泛化能力，所以采用相同的数据集进行模型的构建与评估是不合理的，这样会严重高估模型的准确率，没有办法衡量模型的泛化能力。所以可以初步把数据集划分为**训练集**与**测试集**，使用训练集进行训练，使用测试集进行模型的评估。模型评估方法是对数据集 D 如何划分为训练集 S 和测试集 T 的方法。为了得到更加准确的测试结果，测试集应该与训练集互斥，即测试样本不在训练集中出现，未在训练过程中使用过。目前常见的方法有**留出法**(Hold-Out)、**交叉验证法**(Cross Validation)以及**自助法**(Bootstrap)。

(1) 留出法：将数据集 D 划分为两个互斥的集合，其中一个集合作为训练集 S，另一个集合作为测试集 T。在 S 上训练出模型后，用 T 来评估其测试误差，作为对泛化误差的估计。训练集和测试集的划分要尽可能保持数据分布的一致性，避免因数据划分过程引入额外的偏差而对最终结果产生影响。例如如果存在数据集 D，其中包含 600 个正样本、400 个负样本，数据集 D 划分为 70%样本的训练集和 30%样本的测试集。为了保证训练和测试正负样本的比例与数据 D 比例相同，采用分层抽样的方法。先从 600 个正样本随机抽取 420 次，从 400 个负样本随机抽取 280 次，然后剩下的样本集作为测试集，分层抽样保证了训练集的正负样本的比例与数据集 D 的正负样本比例相同。

（2）交叉验证法：采用留出法实现样本的不同划分方式会导致模型评估的相应结果也会有差别，会使得对模型的评估存在误差。交叉验证法先将数据集 D 划分为 k 个大小相似的互斥子集，每个子集 D_i 通过分层采样得到（如留出法所述，保证正负样本的比例与数据集 D 的比例相同）。然后用 $k-1$ 个子集的并集作为训练集，余下的子集作为测试集；这样就获得 k 组训练/测试集。从而进行 k 次训练和测试，最终返回的是这 k 个测试结果的均值。通常把交叉验证法称为 k 折交叉验证法，k 最常用的取值是 10，此时称为 10 折交叉验证；如果 k 取值为 1，则交叉验证退化为留出法。如果 D 中包含 m 个样本，k 取值为 m 时，则每个子集只有一个样本数据，得到了交叉验证法的一个特例，即**留一法**（Leave-One-Out，LOO）。尽管这种方法可以非常接近准确评估，但是数据集较大时，训练 m 个模型计算开销太大。

（3）自助法：人们希望评估的是用原始数据集 D 训练出的模型，但是留出法和交叉验证法训练的数据集比原始的数据集 D 小，这必然会引入因训练数据集不同导致的估计偏差，所以引入了自助法进行模型评估。自助法是有放回抽样，给定包含 m 个样本的数据集 D，对它进行采样产生数据集 D'；每次有放回地随机从 D 中挑选一个样本，将该样本复制并放入 D'；重复执行 m 次，就得到了包含 m 个样本的数据集 D'，这就是自助法采样的结果。初始数据集 D 中有一部分样本会在数据集 D' 中多次出现，也有一部分样本不会在数据集 D' 中出现。通过自助法采样，初始数据集 D 中约有 36.8%的样本未出现在采样数据集 D' 中，于是可将 D' 用作训练集，我们仍有数据总量约 1/3 的、没在训练集中出现的样本作为测试集用于测试。

留出法、交叉验证法、自助法的对比如表 5-1 所示。

表 5-1　模型评估方法对比

方法名	特　点					
	采样方法	与原始数据分布是否相同	相比原始数据集的容量	是否适用小数据集	是否适用大数据集	是否存在估计偏差
留出法	分层抽样	否	变小	否	是	是
交叉验证法	分层抽样	否	变小	否	是	是
自助法	放回抽样	否	不变	是	否	是

对机器学习的泛化性能进行评估不仅需要有效可行的模型评估方法，还需要有权衡模型泛化性能的评价标准，这就是性能度量（Performance Measure）。性能度量反映了需求，在对比不同的模型时需要采用不同的性能度量指标。例如分类时常用的精度、查准率、查全率，回归时常用的均方误差、均方根误差等。具体性能度量指标将在 5.3 节和 5.4 节介绍。

在分类任务中，经常衡量模型的**精度**（Accuracy），即正确分类与全部分类数据的比值，与之对应，衡量错误分类数据在全部数据所占的比例叫作**错误率**（Error Rate）。错误率与精度是衡量模型性能的最常用的方式，但在特定任务中还需要额外的度量方式。例如在垃圾邮件分类任务中，如果目标分别是“将所有的垃圾邮件选取出来”以及“选取出来的都是垃圾邮件”两类任务，采用精度很难衡量，所以引入了**查准率**（Precision）与**查全率**

(Recall)两个概念。为了更好地介绍查准率与查全率,以二分类问题为例,将分类器预测结果分为以下 4 种情况。

真正(True Positive, TP):被模型预测为正的正样本。

假正(False Positive, FP):被模型预测为正的负样本。

假负(False Negative, FN):被模型预测为负的正样本。

真负(True Negative, TN):被模型预测为负的负样本。

以上 4 种情况组成表 5-2,称为**混淆矩阵**(Confusion Matrix),它是一种特定的矩阵,用来呈现算法性能的可视化效果,每一列代表预测值,每一行代表的是实际的类别。

表 5-2 混淆矩阵

真实数据	预测结果	
	正样本	负样本
正样本	TP	FN
负样本	FP	TN

由表 5-2 可得,精度表示正确分类的测试实例的个数占测试实例总数的比例,计算公式为

$$\text{accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FN} + \text{FP} + \text{TN}) \tag{5-1}$$

查准率针对预测正确的正样本而不是所有预测正确的样本,表示正确分类的正例个数占分类为正例的实例个数的比例,其计算公式为

$$\text{precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{5-2}$$

查全率,也称为召回率,表示正确分类的正例个数占实际正例个数的比例,其计算公式为

$$\text{recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{5-3}$$

在实际应用中,例如在商品推荐系统中,为了尽可能少打扰用户,更希望推荐内容确实是用户感兴趣的,此时查准率更重要;而在逃犯信息检索系统中,更希望尽可能少漏掉逃犯,此时查全率更重要。

查准率和查全率是“鱼”与“熊掌”的关系,通常来讲,查准率高时,查全率往往偏低;而查全率高时,查准率往往偏低。以垃圾邮件分类为例,如果想将垃圾邮件都选取出来,可以将所有邮件都标记为垃圾邮件,则查全率为 1,但这样查准率就会比较低;如果希望垃圾邮件分类模型的查准率足够高,那么可以让分类器挑选最有可能是垃圾邮件的邮件,但这样往往会有大量的垃圾邮件被误识别为正常邮件,此时查全率就会比较低。所以又引入一种新的性能度量指标,称为 **F1 度量**(F1 Score),F1 度量的由来是加权调和平均,更接近于两个数较小的那个,所以查准率和查全率接近时,F1 值最大。F1 度量的定义为

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{5-4}$$

除以上指标外,分类性能度量还有 P-R 曲线、平衡点、ROC 曲线以及 AUC 等指标,这里不再赘述。

下面介绍回归任务的性能度量标准。回归任务的目标是，通过给定数据集 $D=\{x_1,x_2,\cdots,x_m\}$ 预测对应的标签 $\hat{Y}=\{\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_m\}$，使其预测结果 $\hat{Y}$ 尽可能接近数据的标签 $Y=\{y_1,y_2,\cdots,y_m\}$。下面介绍几种常用的性能度量标准。

平均绝对误差(Mean Absolute Error，MAE)：MAE 又称为 $L1$ 范数损失，其衡量预测值与观察值之间的绝对误差的平均值，其公式如下：

$$\mathrm{MAE}=\frac{\sum_{i=1}^{n}|y_i-\hat{y}_i|}{n} \tag{5-5}$$

均方误差(Mean Squared Error，MSE)：MSE 又称为 $L2$ 范数损失，表示预测值与观察值之间的误差平方的平均值，其公式如下：

$$\mathrm{MSE}=\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{n} \tag{5-6}$$

均方根误差(RMSE)，其公式如下：

$$\mathrm{RMSE}=\sqrt{\mathrm{MSE}} \tag{5-7}$$

以上 3 种回归评价指标的取值大小与应用场景有关，很难定义统一的规则评判模型的好坏。下面引入决定系数的概念，其类似于分类中的评价指标，取值范围为 0～1，在不同的应用场景下都可以使用这一评价标准。

决定系数 $\boldsymbol{R}^2$(R-Square)：决定系数中分母为标签 Y 的方差，分子为 MSE。可以根据决定系数的取值，判断模型性能的好坏。$\boldsymbol{R}^2$ 的取值范围为[0，1]。如果模型的决定系数为 0，表示模型的拟合效果很差，$\boldsymbol{R}^2$ 取值越大，说明模型的拟合效果越好。如果结果为 1，说明拟合曲线无错误。其定义为

$$\boldsymbol{R}^2=1-\frac{\sum_{i}(\hat{y}_i-y_i)^2}{\sum_{i}(\bar{y}_i-y_i)^2} \tag{5-8}$$

针对随着样本数量的增加 $\boldsymbol{R}^2$ 值会随之增加，无法定量地说明准确程度的问题，引入了**校正决定系数**(Adjusted R^2)的概念，其抵消了样本数量对 $\boldsymbol{R}^2$ 的影响，可以定量地说明准确程度。公式定义为

$$\text{Adjusted }\boldsymbol{R}^2=1-\frac{(1-R^2)(n-1)}{n-p-1} \tag{5-9}$$

其中，n 为样本数量，p 为特征数量。

聚类也有性能度量标准，聚类性能度量也称为聚类有效性指标，用来评估聚类结果的好坏。聚类结果的簇内相似度越高且保证簇间相似度越低，则认为聚类的性能越好。聚类性能度量分为**外部指标**与**内部指标**两类。

聚类的外部指标是将聚类结果与某个“参考模型”进行比较，例如与领域专家的划分结果进行比较(类似对数据进行标记)。默认参考模型的性能指标是对样本的最优划分，度量的目的就是使聚类结果与参考模型尽可能相近。核心思想是聚类结果中被划分在同一簇样本与参考模型样本也被同样划分到一个簇的概率越高越好。常用的外部指标有

Jaccard 系数、**FM 指数**、**Rand 指数**。

对于给定数据集 $D=\{x_1,x_2,\cdots,x_n\}$，经过聚类算法划分的簇为 $C=\{C_1,C_2,\cdots,C_k\}$，参考模型给出的簇划分为 $C^*=\{C_1^*,C_2^*,\cdots,C_s^*\}$。同时令 $\boldsymbol{l}$ 与 $\boldsymbol{l}^*$ 分别表示数据在 C 与 C^* 中的簇标记向量。将样本两两配对考虑，定义：

$$a=|\ \mathrm{SS}\ |,\mathrm{SS}=\{(x_i,x_j)\mid l_i=l_j,l_i^*=l_j^*,i<j\} \tag{5-10}$$

$$b=|\ \mathrm{SD}\ |,\mathrm{SD}=\{(x_i,x_j)\mid l_i=l_j,l_i^*\neq l_j^*,i<j\} \tag{5-11}$$

$$c=|\ \mathrm{DS}\ |,\mathrm{DS}=\{(x_i,x_j)\mid l_i\neq l_j,l_i^*=l_j^*,i<j\} \tag{5-12}$$

$$d=|\ \mathrm{DD}\ |,\mathrm{DD}=\{(x_i,x_j)\mid l_i\neq l_j,l_i^*\neq l_j^*,i<j\} \tag{5-13}$$

其中，S 表示数据隶属相同簇，D 表示数据隶属不同簇，集合 SS 表示在 $\boldsymbol{C}$ 中隶属相同簇并且在 $\boldsymbol{C}^*$ 中仍隶属相同簇的样本对，a 为集合 SS 中样本对的个数。由于每对样本只可能出现在 4 个集合的其中一个，所以 $a+b+c+d=n(n-1)/2$。

Jaccard 系数(Jaccard Coefficient，JC)定义如下：

$$\mathrm{JC}=\frac{a}{a+b+c} \tag{5-14}$$

FM 指数(Fowlkes and Mallows Index，FMI)定义如下：

$$\mathrm{FMI}=\sqrt{\frac{a}{a+b}\cdot\frac{a}{a+c}} \tag{5-15}$$

Rand 指数(Rand Index，RI)定义如下：

$$\mathrm{RI}=\frac{2(a+d)}{n(n-1)} \tag{5-16}$$

以上性能度量指标的取值均在区间[0，1]内，取值越大代表聚类效果越好。

聚类的内部指标是直接考察聚类结果而不利用参考模型，通过计算簇内样本间的距离以及簇间样本的距离评估模型的性能。核心思想是用簇内样本间距离模拟簇内相似度，簇间样本距离模拟簇间相似度，通过计算距离构建性能指标。常用的内部指标有 **DB 指数**和 **Dunn 指数**。

对于给定数据集 $D=\{x_1,x_2,\cdots,x_n\}$，经过聚类算法划分的簇为 $C=\{C_1,C_2,\cdots,C_k\}$，定义：

$$\mathrm{avg}(C)=\frac{2}{|\ C\ |\ (|\ C\ |-1)}\sum_{1\leqslant i<j\leqslant|C|}\mathrm{dist}(x_i,x_j) \tag{5-17}$$

$$\mathrm{diam}(C)=\max_{1\leqslant i<j\leqslant|C|}\mathrm{dist}(x_i,x_j) \tag{5-18}$$

$$d_{\min}(C_i,C_j)=\min_{x_i\in C_i,x_j\in C_j}\mathrm{dist}(x_i,x_j) \tag{5-19}$$

$$d_{\mathrm{cen}}(C_i,C_j)=\mathrm{dist}(u_i,u_j) \tag{5-20}$$

其中 dist(，) 用于计算两个样本之间的距离；u_i 代表簇 C_i 的中心点 $u_i=\frac{1}{|\ C_i\ |}\sum_{1\leqslant i\leqslant|C_i|}x_i$；avg($C$) 对应簇内样本间的平均距离；diam($C$) 对应于该簇内样本间的最远距离；$d_{\min}(C_i,C_j)$ 对应两簇间样本的最近距离；$d_{\mathrm{cen}}(C_i,C_j)$ 对应两簇中心点的距离。基于以上指标可得以下常用的聚类性能度量内部指标，其中 DB 指数值越小越好，Dunn 指数值越大越好。

DB 指数(Davies-Bouldin Index,DBI)定义如下：

$$\mathrm{DBI}=\frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\mathrm{avg}(C_i)+\mathrm{avg}(C_j)}{d_{\mathrm{cen}}(C_i,C_j)}\right) \tag{5-21}$$

Dunn 指数(Dunn Index,DI)定义如下：

$$\mathrm{DI}=\min_{1\leqslant i\leqslant k}\left\{\min_{j\neq i}\left(\frac{d_{\min}(C_i,C_j)}{\max\limits_{1\leqslant x\leqslant k}\mathrm{diam}(C_x)}\right)\right. \tag{5-22}$$

5.1.4　发展历程

人工智能(Artificial Intelligence),英文缩写为 AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能是计算机科学的一个分支,它企图了解智能的实质,并生产出一种新的能以人类智能相似的方式做出反应的智能机器,该领域的研究包括机器人、语言识别、图像识别、自然语言处理和专家系统等。人工智能可以对人的意识、思维的信息过程进行模拟。人工智能不是人的智能,但能像人那样思考,也可能超过人的智能。人工智能的发展历程大致如下：

第一阶段：20 世纪 50 年代中叶到 20 世纪 60 年代初,人工智能研究进入“推理期”,认为只需要为机器赋予逻辑推理能力,机器就会拥有智能。

第二阶段：20 世纪 60 年代中叶到 20 世纪 70 年代初,人工智能研究进入冷静时期,人们发现仅有逻辑推理能力无法使机器具有智能。

第三阶段：20 世纪 70 年代中叶到 20 世纪 80 年代初,人工智能进入“知识期”,人们提出要想使机器拥有智能,必须设法让机器拥有知识,这个阶段大量的专家系统的问世获得了较多的应用领域的成果。

第四阶段：20 世纪 80 年代中叶到现在,人工智能进入“学习期”,由于专家系统面临着需要人为将知识总结出来教给计算机的困难,于是有学者想到应该让机器能够自主学习知识,机器学习正式走入人工智能舞台[1]。

机器学习是人工智能的核心,是使计算机拥有智能的根本途径,机器学习的发展是整个人工智能发展史上颇为重要的一个分支。从 1952 年 IBM 科学家亚瑟·塞缪尔研制了一个西洋跳棋程序开始,机器学习进入研究者的视野之内;1957 年,罗森·布拉特(F. Rosenblatt)[10]提出了感知机(Perceptron)模型,成功处理了线性分类问题,为现在的神经网络以及深度学习开创了基础。

1967 年,最近邻算法(The Nearest Neighbor Algorithm)[11]出现,这是一种基于模板匹配思想的算法,虽然简单,但很有效,至今仍在被使用。同年,K 均值算法[12]也被提出,此后出现了其大量的改进算法,取得了成功的应用,是所有聚类算法中变种和改进型最多的。1969 年,马文·明斯基将感知器热度推到顶峰,他提出了著名的 XOR 问题和感知器数据线性不可分的情形。

1980 年,在卡内基-梅隆大学(CMU)召开了第一届机器学习国际研讨会,标志着机器学习研究已在全世界兴起。1981 年,多层感知器(MLP)在伟博斯的神经网络反向传播(BP)算法中具体提出。BP 仍然是今天神经网络架构的关键因素。1986 年诞生了用于训练多层神经网络的真正意义上的反向传播算法,这是现在的深度学习中仍然被使用的训

练算法，奠定了神经网络走向完善和应用的基础。1986 年，昆兰提出 ID3[13]决策树算法，虽然简单，但可解释性强，这使得决策树至今在一些问题上仍被使用。1989 年，LeCun[14]设计出了第一个真正意义上的卷积神经网络，用于手写数字的识别，这是现在被广泛使用的深度卷积神经网络的鼻祖。

1995 年，支持向量机（Support Vector Machines，SVM）[15]由瓦普尼克和科尔特斯在大量理论和实证的条件下提出。从此将机器学习社区分为神经网络社区和支持向量机社区。2006 年，神经网络研究领域领军者 Hinton 提出了神经网络 Deep Learning 算法，使神经网络的能力大大提高，向支持向量机发出挑战，开启了深度学习在学术界和工业界的研究和应用浪潮。

5.2 Sklearn 库基本使用

5.2.1 Sklearn 库简介

自 2007 年发布以来，Scikit-learn[8]已经成为 Python 重要的机器学习库了。Scikit-learn 简称 Sklearn，支持包括分类、回归、降维和聚类四大机器学习算法。还包含了特征提取、数据预处理和模型评估三大模块。Sklearn 是 SciPy 的扩展，建立在 NumPy 和 Matplotlib 库的基础上。利用这几大模块的优势，可以大大提高机器学习的效率。

Sklearn 拥有完善的文档，上手容易，具有丰富的 API，在学术界颇受欢迎。Sklearn 已经封装了大量的机器学习算法，包括 LIBSVM 和 LIBINEAR。同时 Sklearn 内置大量数据集，节省了获取和整理数据集的时间。

Sklearn 软件包支持主流的有监督机器学习方法（Supervised Machine Learning Algorithm）、无监督机器学习方法（Unsupervised Machine Learning Algorithm）。有监督的机器学习方法包括通用的线性模型、支持向量机、决策树、贝叶斯方法等。无监督的机器学习方法包括聚类、因子分析、主成分分析、无监督神经网络等。

目前 Sklearn 的版本为 0.21，安装需要 Python 版本在 3.5 及以上，NumPy 版本在 1.11.0 及以上，SciPy 版本在 0.17 及以上。如果采用 Python 2.7 版本，则可以使用 Sklearn 0.20 版本。Sklearn 支持 pip 安装以及 conda 安装。

（1）pip 命令安装：pip install -u scikit-learn。

（2）conda 命令安装：conda install scikit-learn。

5.2.2 基本使用介绍

传统的机器学习任务从开始到建模的一般流程就：数据获取→数据预处理→模型训练→模型预测与评估→保存。本文按照传统机器学习的流程，总结每一步流程中都有哪些常用的函数以及它们的用法是怎么样的。

1. 数据获取

Sklearn 中包含了大量的优质的数据集，在机器学习的过程中，可以使用这些数据集

实现不同的模型，从而提高动手实践能力，同时这个过程也可以加深对理论知识的理解和把握。

Sklearn 的 datasets 模块中包含有许多小数据集，例如鸢尾花数据集(iris)、波士顿数据集(boston)、手写数字数据集(digits)，这些数据集已经存在于 Sklearn 库中，可以直接导入；datasets 还包含有部分真实数据集，例如新闻分类数据集、人脸数据集等，此类数据集在第一次导入时会自动下载，之后就可以直接使用。Sklearn 库导入数据集如下所示。

【例 5-1】　Sklearn 库导入数据集。

要使用 Sklearn 中的数据集，必须导入 datasets 模块。使用 dir 函数给出这个模块下的函数列表。

```
In[1]: from sklearn import datasets
       dir(datasets)
```

鸢尾花(iris)数据集中包含三类鸢尾花数据，是常用的分类实验数据集。下面代码表示导入鸢尾花数据集。

```
In[2]: Iris = datasets.load_iris()        #导入数据集
       data = Iris.data                   #获取样本特征向量
       target = Iris.target               #获得样本 label
       print (data,target)
```

数据集中的样本的特征向量以及样本标签的存储格式均为矩阵。可以通过下面的代码打印查看。

```
In[3]: print (type(data),type(target))
Out[3]: <class 'numpy.ndarray'>
        <class 'numpy.ndarray'>
```

数据集中包含三种鸢尾花，分别是山鸢尾(setossa)、变色鸢尾(versicolor)和弗吉尼亚鸢尾(virginica)。可以通过以下代码获取对应名称。

```
In[4]: print(Iris.target_names)
Out[4]: ['setosa' 'versicolor' 'virginica']
```

数据集中的样本包含 4 个属性特征，分别为花萼长度、花萼宽度、花瓣长度、花瓣宽度。可通过以下代码查看对应属性。

```
In[5]: print(Iris.feature_names)
Out[5]: ['sepal length (cm)', 'sepal width (cm)',
         'petal length (cm)', 'petal width (cm)']
```

数据集三类数据每类包含 50 个样本，共 150 个样本。每个样本包含 4 个特征向量和 1 个类别向量(label)。