机器学习综述

# 机器学习的历史

## 基础奠定的热烈时期

1949年，Hebb根据神经心理学的学习机制开始了机器学习的第一步。从此被称为Hebb学习规则。Hebb学习规则是一种无监督的学习规则。这种学习的结果是使网络能够提取训练集的统计特征，从而根据相似性将输入信息划分为若干类别。这与人类观察和理解世界的过程非常一致。人类对世界的观察和理解在很大程度上是根据事物的统计特征来分类的。

从上面的公式可以看出，权重调整量与输入和输出的乘积成正比。显然，频繁出现的模式会对权重向量产生很大影响。在这种情况下，Hebb学习规则需要预先设置权重饱和值，以防止在输入和输出的正负值始终相同时权重的无约束增长。

1950年，艾伦·图灵创建了图灵测试，以确定计算机是否智能。根据图灵测试，如果一台机器可以与人类对话而不被识别为机器，那么它就是智能的。这种简化使图灵能够令人信服地说明“思维机器”是可能的。

1952年，IBM科学家亚瑟·塞缪尔（Arthur Samuel）开发了一个跳棋程序。该程序可以观察当前位置并学习隐式模型，从而为后续行动提供更好的指导。塞缪尔发现，随着游戏程序运行时间的增加，它可以实现越来越好的后续指导。

通过这个项目，塞缪尔驳斥了普罗维登斯的模型，即机器不能超越人类，不能像人类一样编写代码和学习。他创建了“机器学习”，并将其定义为“一个不需要显式编程就能提供计算机能力的研究领域”。

1957年，Rosen Blatt以神经感知科学为背景提出了第二个模型，该模型与当今的机器学习模型非常相似。这在当时是一个非常激动人心的发现，它比赫布的想法更适用。基于这个模型，罗森·布拉特设计了第一个计算机神经网络——感知器，它模拟人脑的运作。

1960年，vidro首次将增量学习规则用于感知器训练步骤。这种方法后来被称为最小二乘法。两者的结合创造了一个好的线性分类器。

1967年，最近邻算法出现，使计算机可以进行简单的模式识别。KNN算法的核心思想是，如果特征空间中的k个最近样本中的大多数属于一个类别，则该样本也属于该类别，并且具有该类别中样本的特征。在确定分类决策时，该方法仅根据最近的一个或多个样本的类别确定要划分的样本类别。

1969年，马文·明斯基将感知器的兴奋推到了最高峰。他提出了著名的异或问题和感知器数据线性不可分割的情况。明斯基还将人工智能技术与机器人技术相结合，开发出世界上第一台能够模拟人类活动的机器人RobotC，将机器人技术提升到了一个新的水平。明斯基的另一个重大举措是成立著名的“思维机器公司”，开发智能计算机。

## 停滞不前的冷静时期

从20世纪60年代中期到70年代末，机器学习的发展几乎处于停滞状态。尽管温斯顿的结构学习系统和海斯·罗斯的基于逻辑的归纳学习系统在这一时期取得了巨大的进步，但它们只能学习一个概念，不能应用于实践。此外，由于理论上的缺陷，神经网络学习机未能达到预期的效果，陷入了低潮。

这一阶段的研究目标是模拟人类的概念学习过程，并使用逻辑结构或图形结构作为机器的内部描述。机器可以使用符号来描述概念（符号概念获取），并提出关于学习概念的各种假设。

## 重拾希望的复兴时期

自20世纪70年代末以来，人们从学习单个概念扩展到学习多个概念，探索不同的学习策略和学习方法。在此期间，机器学习在大量时间应用中重新引起人们的注意，并慢慢恢复。

1980年，第一届机器学习国际研讨会在美国卡内基梅隆大学（CMU）举行，标志着机器学习研究在全世界的兴起。从那时起，机器归纳学习进入应用。

经过一些挫折，多层感知器（MLP）于1981年由伟博斯在神经网络反向传播（BP）算法中特别提出。当然，BP仍然是当今神经网络体系结构中的一个关键因素。有了这些新的思路，神经网络的研究就加快了。

从1985年到1986年，神经网络研究人员（Rumelhart、Sinton、Williams he、Nelson）先后提出了MLP与BP训练相结合的思想。

昆兰在1986年提出了一个非常著名的ML算法。我们称之为决策树算法，更准确地说是ID3算法。这是主流机器学习的另一个火花。此外，与黑盒神经网络模型不同，决策树ID3算法也被用作软件。通过使用简单的规则和清晰的引用，可以找到更多的实际应用场景。

## 现代机器学习的成型时期

1990年,Schapire最先构造出一种多项式级的算法,对该问题做了肯定的证明,这就是最初的Boosting算法。一年后,Freund提出了一种效率更高的Boosting算法。但是,这两种算法存在共同的实践上的缺陷,那就是都要求事先知道弱学习算法学习正确的下限。

1995年,Freund和schapire改进了Boosting算法,提出了AdaBoost(AdaptiveBoosting)算法,该算法效率和Freund于1991年提出的Boosting算法几乎相同,但不需要任何关于弱学习器的先验知识,因而更容易应用到实际问题当中。

同年，Vapnik和Cortes在大量理论和经验条件下提出了机器学习领域最重要的突破之一——支持向量机。此后，机器学习社区又分为神经网络社区和支持向量机社区。

支持向量机、boosting、最大熵法等。这些模型的结构基本上可以看作是一层隐藏节点或无隐藏节点（如LR）。这些模型在理论分析和应用上都取得了很大的成功。

2001年，Breman博士提出了另一个综合决策树模型。它由一个随机子集的实例组成，每个节点从一系列随机子集中选择。由于这种特性，它被称为随机森林（RF）。随机森林也从理论和经验上证明了其抵抗过度拟合的能力。

甚至连AdaBoost算法在数据过拟合和离群实例中都表现出了弱点，而随机森林是针对这些警告更稳健的模型。随机森林在许多不同的任务，像DataCastle、Kaggle等比赛等都表现出了成功的一面。

## 大放光芒的蓬勃发展时期

在机器学习发展分为两个部分，浅层学习（ShallowLearning）和深度学习（DeepLearning）。浅层学习起源上世纪20年代人工神经网络的反向传播算法（Back-propagation）的发明，使得基于统计的机器学习算法大行其道，虽然这时候的人工神经网络算法也被称为多层感知机（MultiplelayerPerception），但由于多层网络训练困难，通常都是只有一层隐含层的浅层模型。

神经网络研究领域的领军人物Hinton于2006年提出了神经网络深度学习算法，极大地提高了神经网络的能力，对支持向量机提出了挑战。2006年，机器学习领域的领导者Hinton和他的学生salakhutdinov在顶级学术期刊《Scince》上发表了一篇文章，开启了学术界和工业界的深度学习浪潮。

Hinton的学生YannLeCun的LeNets深度学习网络可以被广泛应用在全球的ATM机和银行之中。同时，YannLeCun和吴恩达等认为卷积神经网络允许人工神经网络能够快速训练，因为其所占用的内存非常小，无须在图像上的每一个位置上都单独存储滤镜，因此非常适合构建可扩展的深度网络，卷积神经网络因此非常适合识别模型。

2015年，为纪念人工智能概念提出60周年，LeCun、Bengio和Hinton推出了深度学习的联合综述。

目前，统计学习领域最流行的方法是深度学习和支持向量机（SVM），它们是统计学习的代表性方法。

神经网络和支持向量机一直处于“竞争”关系。支持向量机在不知道非线性映射显式表达式的情况下应用核函数展开定理；由于线性学习机是建立在高维特征空间中的，与线性模型相比，它不仅几乎不增加计算复杂度，而且在一定程度上避免了维数灾难。以往的神经网络算法易于训练，需要设置大量的经验参数；训练速度相对较慢，当级别较少（小于或等于3）时，效果并不比其他方法好。

神经网络模型似乎能够实现比较困难的任务，如目标识别、语音识别、自然语言处理等。然而，需要注意的是，这并不意味着其他机器学习方法的终结。虽然深度学习的成功案例增长迅速，但这些模型的训练成本相当高，而且调整外部参数也非常麻烦。同时，支持向量机的简单性使得它仍然是应用最广泛的机器学习方法。

# 机器学习类别

按照学习态度和不同的灵感来源，可将机器学习大致分为5个具有不同思想的类别，包括符号主义（Symbol⁃ists）、联结主义（Connectionists）、进化主义（Evolutionaries）、贝叶斯主义（Bayesians）和类推主义（Analogizer）。每个类别围绕各自的基本思想，有重点研究领域及相应算法。

符号主义认为，所有信息均可简化为操作符号，符号学者使用符号、规则和逻辑表征知识与进化逻辑进行推理，其主算法是逆向演绎，包括规则学习和决策树等。

联结主义认为学习是大脑所做的事情，因此联结学者使用概率矩阵和加权神经元动态地识别和归纳模式，从而对大脑进行逆向演绎，其主算法是反向传播学习算法，如神经网络。

进化主义从对自然选择的解释中发现了学习本质，它通过生成变化，再根据特定目标获取最优者，在计算机中实现对自然的模仿，其主算法是基因编程，如遗传算法。

贝叶斯主义关注的中心是不确定性，其认为从知识获取过程到一切既定知识均具备不确定性。贝叶斯学者利用事件发生的概率大小进行计算，其主算法是贝叶斯定理，如朴素贝叶斯和马尔可夫链。

类推主义从关系间的相似性着手，推导出其它关系。类推学者根据约束条件优化函数，通过找出需被记忆的经历并弄清之间的结合关系，实现新场景预测，其主算法是支持向量机。

假设要解决的新问题存储在数据库中，而现有数据库非常大，则出现此问题的概率非常低。记忆不能用作学习算法。符号主义的代表人物之一汤姆·米切尔（Tom Mitchell）意识到，机器学习必须预先设定这个概念，即引入额外的假设，以便在能力范围内总结出最普遍的规则。另一个问题是如何合理准确地概括已知甚至未知的知识？各种机器学习算法正试图解决这一归纳困难。Ryszard Michalski是象征主义的创始人之一，他将影响结果的任何因素与心理学中的“合取概念”结合起来。他设想，在机器学习任何规则后，下一条规则将包含尽可能多的剩余解，直到获得与该解对应的完整规则集。然而，大多数算法掌握的知识有限，这种“分而治之”的归纳算法容易导致过拟合问题。在这方面，斯蒂芬·马格尔顿和其他人借鉴了“归纳就是反向演绎”的思想，并于1988年设计了第一个归纳逻辑程序。面对海量数据，符号学家提出了决策树归纳法，通过对规则的排序或选择，解决了多个概念规则对应一个实例的问题，从而保证了实例与规则的一一对应，使决策树准确易懂。rossquinlan的研究使其成为分类问题的领导者。20世纪80年代，符号学迅速占据了机器学习的主导地位，在专家系统的应用中取得了巨大的成功，但由于复杂的规则编码，符号学逐渐消失。随着人工智能技术的进步，上述先驱的工作之间出现了新的联系，符号学机器学习再次活跃起来。

连接主义认为，仅仅通过逻辑规则来定义概念是不够的。连接主义相信赫布理论，即“同时激活的神经元将被连接”，该理论反过来分析人脑的运作。大约在1960年，弗兰克·罗森布拉特通过给麦卡洛赫·皮特神经元之间的连接赋予不同的权重来设计感知机。感知器结构简单，仅通过示例训练就可以区分图像和声音。然而，如果正实例和负实例不能被超平面（即典型的排除或函数）分开，感知器就无法学习。直到1985年，同时具有感官和隐藏神经元的玻尔兹曼机器的诞生，将联结主义的复兴推向了顶峰。虽然Boltzmann机器有效地解决了表扬分配问题，但在实践过程中学习行为非常缓慢和困难。不久之后，Yannlecn发现了一种反向传播算法，该算法不仅可以学习XOR，还可以有效地处理赞扬分布问题，并与YoshuaBengio等人一起不断优化。叠加自动编码器和卷积神经网络等深度学习算法的诞生曾使联结主义成为机器学习的主导思想。

进化主义和联结主义一样，都试图效仿自然进行学习的方法。JohnHolland最早研究的是神经网络，但随着关于进化的数学理论体系逐渐成熟，其关注点转变为一种从达尔文自然选择理论转化的算法。遗传算法类似选择育种，它通过模拟点突变和染色体交叉过程生成变化，然后引入适应度函数给程序和目标的契合度打分。在解决垃圾邮件过滤的应用过程中，他提出分类器系统的规则集，并利用桶队算法处理其面临的赞誉分布问题。尽管如此，与分层感知器相比，分类器系统可适用的领域十分有限。1987年JohnKoza发现了基因编程方法，即对成熟的计算机程序自身进行进化。这种对程序树而非字符串进行交叉的方法，使学习活动变得更为灵活。目前联结主义最引人关注的应用是具有自我意识和创造力进化的机器人研发。反观进化主义和联结主义主算法的不同侧重点，前者注重结构学习，而后者则通过权值学习解决大部分工作。借助遗传算法寻找神经网络最优结构将成为两种思想的融合点。

与模仿自然的方法不同，贝叶斯主义和符号主义都试图从基本原理出发探索算法学习的方法。基于贝叶斯定理的贝叶斯理论发明了朴素贝叶斯分类器，由于能够获得输入和输出之间的两两相关性而得到广泛应用。最成功的案例之一是David heckerman的垃圾邮件过滤器。朴素贝叶斯方法和马尔可夫链都是贝叶斯网络的特例。后者是朱迪亚·珀尔（Judea pearl）在20世纪80年代创建的关系图，其中包含任意结构特征，并且允许特征之间的干涉。然后是使变量呈指数增长的推理问题。Pearl&Jordan提出了“环路信念传播”的思想和优化易于处理的近似推理内部参数的方法。

上述四个类别的一个共同缺点是，在数据不足的情况下，他们学习研究的显式模型无法继续有效。然而，类推主义可以只能从小数据中学习，包括高效的最近邻算法和精确的支持向量机。彼得·哈特（Peter Hart）提出的最近邻算法也被认为是一种惰性学习算法。它将每个数据点转化为一个微分类器，每次只构建一个局部模型。它的优点是学习过程简单、快速，但与其他学习算法相比，它受维数灾难的影响更大。直到20世纪90年代，Vladimir Vapnik等人开发的支持向量机成为类比主义的新代表。支持向量机与加权k近邻算法非常相似，但前者可以提供平滑的边界，而不需要过拟合。道格拉斯·霍夫斯塔特高度评价类比推理。

# 机器学习算法

## 决策树

决策树是符号主义的主算法。决策树作为模仿人脑在日常生活中处理决策问题的方法，具有面对“是否”、“好坏”等二分类任务的二叉树结构。在得到问题结论的过程中，所有对数据中各特征子决策判断的累积，使求解范围不断缩小。一棵完整的决策树由一个包含数据和特征全集的根节点、若干代表特征判定的过程节点以及相应代表决策结果的叶节点组成。从根节点到叶节点的过程参照“分而治之”的机制展开。

**决策树学习的核心问题之一是特征划分。3种经典划分方法形成了决策树的3种代表性算法：ID3、C4.5和CART。**

**ID3算法是最早提出的决策树算法，他就是利用信息增益来选择特征的。**

**C4.5算法是ID3的改进版，他不是直接使用信息增益，而是引入“信息增益比”指标作为特征的选择依据。**

**CART算法使用了基尼系数取代了信息熵模型。**

**剪枝是决策树学习的第二个核心问题。剪枝可以减少决策树的层数，降低运算复杂度，并且可以解决数据的过拟合问题。按照决策树生成与否分为先、后两种剪枝方式。**

### **决策树的学习步骤**

#### 特征选择

特征选择决定了使用哪些特征来做判断。在训练数据集中，每个样本的属性可能有很多个，不同属性的作用有大有小。因而特征选择的作用就是筛选出跟分类结果相关性较高的特征，也就是分类能力较强的特征。

在特征选择中通常使用的准则是信息增益。信息增益依赖于香农提出的信息熵的概念，希望在特征选择后能获得更小的信息熵。

#### 决策树生成

选择好特征后，就从根节点触发，对节点计算所有特征的信息增益，选择信息增益最大的特征作为节点特征，根据该特征的不同取值建立子节点；对每个子节点使用相同的方式生成新的子节点，直到信息增益很小或者没有特征可以选择为止。

#### 决策树剪枝

决策树的剪枝策略最基本的有两种：预剪枝（pre-pruning）和后剪枝（post-pruning）。

预剪枝（pre-pruning）：预剪枝就是在构造决策树的过程中，先对每个结点在划分前进行估计，若果当前结点的划分不能带来决策树模型泛华性能的提升，则不对当前结点进行划分并且将当前结点标记为叶结点。

后剪枝（post-pruning）：后剪枝就是先把整颗决策树构造完毕，然后自底向上的对非叶结点进行考察，若将该结点对应的子树换为叶结点能够带来泛华性能的提升，则把该子树替换为叶结点。

对比预剪枝和后剪枝，能够发现，后剪枝决策树通常比预剪枝决策树保留了更多的分支，一般情形下，后剪枝决策树的欠拟合风险小，泛华性能往往也要优于预剪枝决策树。但后剪枝过程是在构建完全决策树之后进行的，并且要自底向上的对树中的所有非叶结点进行逐一考察，因此其训练时间开销要比未剪枝决策树和预剪枝决策树都大得多。

## **人工神经网络**

联结主义主算法是人工神经网络。人工神经网络的灵感来自其生物学对应物。生物神经网络使大脑能够以复杂的方式处理大量信息。大脑的生物神经网络由大约1000亿个神经元组成，这是大脑的基本处理单元。神经元通过彼此之间巨大的连接来执行其功能。人脑大约有100万亿个突触，每个神经元约有1,000个。

人工神经网络是一种由具有自适应性的简单单元构成的广泛并行互联的网络，它的组织结构能够模拟生物神经系统对真实世界所做出的交互反应。

以神经网络中的最小单元，即经典M-P神经元模型为例，当前神经元收到来自前方神经元输出信号X加上权值ω后的信号，借助响应函数f产生最终输出信号Y，与阈值θ对比。

图示

描述已自动生成

人工神经网络由一个输入层和一个输出层组成，其中输入层从外部源（数据文件，图像，硬件传感器，麦克风等）接收数据，一个或多个隐藏层处理数据，输出层提供一个或多个数据点基于网络的功能。在得到一个训练集之后，它能通过学习提取所观察事物的各个部分的特征，将特征之间用不同网络节点连接，通过训练连接的网络权重，改变每一个连接的强度，直到顶层的输出得到正确的答案。

在神经网络发展过程中出现了前馈神经网络和递归神经网络两种结构。后者打破了前者关于输入和输出相互独立的假设，通过网络建立的环形结构使某些输出信号反馈成为输入信息。

前馈神经网络适合于图像处理，递归神经网络适合于自然语言处理。

误差反向传播（BP）算法既可以用在前馈神经网络，又可以用于递归神经网络训练。

图示

描述已自动生成

单隐藏层神经网络

其中，为输入层节点i和隐含层节点j之间的权值，为节点j和输出层节点k之间的权值。

### 前向传播

刺激由前一层传向下一层，故而称之为前向传递

对于非线性分类问题，逻辑回归会使用多项式扩展特征，导致维度巨大的特征向量出现，而在神经网络中，并不会增加特征的维度，即不会扩展神经网络输入层的规模，而是通过增加隐含层，矫正隐含层中的权值，来不断优化特征，前向传播过程每次在神经元上产出的激励值都可看做是优化后的特征。

手表的卡通人物

描述已自动生成

代价函数为：



L为神经网络总共包含的层数，为第l层神经元数目，K为输出层的神经元数目

### 反向传播

由于神经网络允许多个隐含层，即，各层的神经元都会产出预测，因此，就不能直接利用传统回归问题的梯度下降法来最小化J(Θ) ，而需要逐层考虑预测误差，并且逐层优化。用反向传播法优化预测。首先定义各层的预测误差为向量 δ(l)：



反向传播中的反向二字也正是从该公式中得来，本层的误差 δ(l) 需要由下一层的误差 δ(l+1) 反向推导。

图示

描述已自动生成

## 遗传算法

进化主义主算法是遗传算法。遗传算法是从代表问题可能潜在的解集的一个种群开始的，而一个种群则由经过基因编码的一定数目的个体组成。每个个体实际上是染色体带有特征的实体。

染色体作为遗传物质的主要载体，即多个基因的集合，其内部表现（即基因型）是某种基因组合，它决定了个体的形状的外部表现，如黑头发的特征是由染色体中控制这一特征的某种基因组合决定的。因此，在一开始需要实现从表现型到基因型的映射即编码工作。由于仿照基因编码的工作很复杂，我们往往进行简化，如二进制编码。

初代种群产生之后，按照适者生存和优胜劣汰的原理，逐代演化产生出越来越好的近似解，在每一代，根据问题域中个体的适应度大小选择个体，并借助于自然遗传学的遗传算子进行组合交叉和变异，产生出代表新的解集的种群。

这个过程将导致种群像自然进化一样的后生代种群比前代更加适应于环境，末代种群中的最优个体经过解码，可以作为问题近似最优解。

图示

描述已自动生成

其中迭代终止条件是群体适应度最大值与平均值相差E小于容差ε或迭代次数t超过最大进化代数T，表达个体适应度F和其被选中概率P的常用公式为：

值得注意的是，基因编码方式的合理选择对遗传算法效果影响显著。

文本

描述已自动生成

### 编码方式

#### 二进制编码法

就像人类的基因有AGCT 4种碱基序列一样。不过在这里我们只用了0和1两种碱基,然后将他们串成一条链形成染色体。一个位能表示出2种状态的信息量，因此足够长的二进制染色体便能表示所有的特征，这便是二进制编码。

优点是：

编码、解码操作简单易行

交叉、变异等遗传操作便于实现

符合最小字符集编码原则

利用模式定理对算法进行理论分析

缺点是：

对于一些连续函数的优化问题，由于其随机性使得其局部搜索能力较差

当解迫近于最优解后，由于其变异后表现型变化很大，会远离最优解，达不到稳定

#### 浮点编码法

浮点法是指个体的每个基因值用某一范围内的一个浮点数来表示。在浮点数编码方法中，必须保证基因值在给定的区间限制范围内，遗传算法中所使用的交叉、变异等遗传算子也必须保证其运算结果所产生的新个体的基因值也在这个区间限制范围内。

优点：

适用于在遗传算法中表示范围较大的数

适用于精度要求较高的遗传算法

便于较大空间的遗传搜索

改善了遗传算法的计算复杂性，提高了运算交率

便于遗传算法与经典优化方法的混合使用

便于设计针对问题的专门知识的知识型遗传算子

便于处理复杂的决策变量约束条件

#### 符号编码法

符号编码法是指个体染色体编码串中的基因值取自一个无数值含义、而只有代码含义的符号集

优点：

符合有意义积术块编码原则

便于在遗传算法中利用所求解问题的专门知识

便于遗传算法与相关近似算法之间的混合使用

### 评价适应度

适应度函数也称评价函数，是根据目标函数确定的用于区分群体中个体好坏的标准。适应度函数总是非负的，而目标函数可能有正有负，故需要在目标函数与适应度函数之间进行变换。

### 选择函数

遗传算法中的选择操作就是用来确定如何从父代群体中按某种方法选取那些个体，以便遗传到下一代群体。选择操作用来确定重组或交叉个体，以及被选个体将产生多少个子代个体。

轮盘赌选择：回放式随机采样方法。每个个体进入下一代的概率等于它的适应度值与整个种群中个体适应度值和的比例。选择误差较大。

随机竞争选择：每次按轮盘赌选择一对个体，然后让这两个个体进行竞争，适应度高的被选中，如此反复，直到选满为止。

最佳保留选择：首先按轮盘赌选择方法执行遗传算法的选择操作，然后将当前群体中适应度最高的个体结构完整地复制到下一代群体中。

无回放随机选择：根据每个个体在下一代群体中的生存期望来进行随机选择运算。计算群体中每个个体在下一代群体中的生存期望数目N。若某一个体被选中参与交叉运算，则它在下一代中的生存期望数目减去0.5，若某一个体未 被选中参与交叉运算，则它在下一代中的生存期望数目减去1.0。随着选择过程的进行，若某一个体的生存期望数目小于0时，则该个体就不再有机会被选中。

确定式选择：按照一种确定的方式来进行选择操作。具体操作过程如下：计算群体中各个个体在下一代群体中的期望生存数目N。用N的整数部分确定各个对应个体在下一代群体中的生存数目。用N的小数部分对个体进行降序排列，顺序取前M个个体加入到下一代群体中。至此可完全确定出下一代群体中Ｍ个个体。

无回放余数随机选择：可确保适应度比平均适应度大的一些个体能够被遗传到下一代群体中，因而选择误差比较小。

均匀排序：对群体中的所有个体按期适应度大小进行排序，基于这个排序来分配各个个体被选中的概率。

最佳保存策略：当前群体中适应度最高的个体不参与交叉运算和变异运算，而是用它来代替掉本代群体中经过交叉、变异等操作后所产生的适应度最低的个体。

随机联赛选择：每次选取几个个体中适应度最高的一个个体遗传到下一代群体中。

排挤选择：新生成的子代将代替或排挤相似的旧父代个体，提高群体的多样性。

### 遗传--染色体交叉

遗传算法的交叉操作，是指对两个相互配对的染色体按某种方式相互交换其部分基因，从而形成两个新的个体。

单点交叉：指在个体编码串中只随机设置一个交叉点，然后再该点相互交换两个配对个体的部分染色体。

两点交叉：在个体编码串中随机设置了两个交叉点，然后再进行部分基因交换。

均匀交叉：两个配对个体的每个基因座上的基因都以相同的交叉概率进行交换，从而形成两个新个体。

算术交叉：由两个个体的线性组合而产生出两个新的个体。该操作对象一般是由浮点数编码表示的个体。

### 变异--基因突变

遗传算法中的变异运算，是指将个体染色体编码串中的某些基因座上的基因值用该基因座上的其它等位基因来替换，从而形成新的个体。

基本位变异：对个体编码串中以变异概率、随机指定的某一位或某几位仅因座上的值做变异运算。

均匀变异：分别用符合某一范围内均匀分布的随机数，以某一较小的概率来替换个体编码串中各个基因座上的原有基因值。（特别适用于在算法的初级运行阶段）

边界变异：随机的取基因座上的两个对应边界基因值之一去替代原有基因值。特别适用于最优点位于或接近于可行解的边界时的一类问题。

非均匀变异：对原有的基因值做一随机扰动，以扰动后的结果作为变异后的新基因值。对每个基因座都以相同的概率进行变异运算之后，相当于整个解向量在解空间中作了一次轻微的变动。

高斯近似变异：进行变异操作时用符号均值为Ｐ的平均值，方差为的正态分布的一个随机数来替换原有的基因值。

## 朴素贝叶斯

贝叶斯主义主算法是朴素贝叶斯算法。朴素贝叶斯法是基于贝叶斯定理与特征条件独立假设的分类方法。最为广泛的两种分类模型是决策树模型和朴素贝叶斯模型。和决策树模型相比，朴素贝叶斯分类器(NBC)发源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率。同时，NBC模型所需估计的参数很少，对缺失数据不太敏感，算法也比较简单。理论上，NBC模型与其他分类方法相比具有最小的误差率。但是实际上并非总是如此，这是因为NBC模型假设属性之间相互独立，这个假设在实际应用中往往是不成立的，这给NBC模型的正确分类带来了一定影响。

源于统计学的贝叶斯定理可以表示为：

P（原因|结果）=P（原因）×P（结果|原因）/P（结果）

朴素贝叶斯算法假设所有特征的出现相互独立互不影响，每一特征同等重要，又因为其简单，而且具有很好的可解释性一般。相对于其他精心设计的更复杂的分类算法，朴素贝叶斯分类算法是学习效率和分类效果较好的分类器之一。朴素贝叶斯算法一般应用在文本分类，垃圾邮件的分类，信用评估，钓鱼网站检测等。

面对多分类问题，用训练数据X代替上式“结果”，用离散的数据特征种类C代替“原因”。为方便得到条件概率P(x|c)，忽略各特征间对分类作用的关联，此时有：

手机屏幕截图

中度可信度描述已自动生成

使样本按最低风险选择特征种类的最优分类判定准则有：

文本

描述已自动生成

其中C为特征种类c的取值集合，数量为d个。

但由于实际应用中各特征很可能相互干涉，且训练数据有缺失，于是又从中演化出其它算法，以增强泛化能力。

文本, 信件

描述已自动生成

### 优缺点

优点：

朴素贝叶斯模型发源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率。

对大数量训练和查询时具有较高的速度。即使使用超大规模的训练集，针对每个项目通常也只会有相对较少的特征数，并且对项目的训练和分类也仅仅是特征概率的数学运算而已；

对小规模的数据表现很好，能个处理多分类任务，适合增量式训练（即可以实时的对新增的样本进行训练）；

对缺失数据不太敏感，算法也比较简单，常用于文本分类；

朴素贝叶斯对结果解释容易理解。

缺点：

需要计算先验概率；

分类决策存在错误率；

对输入数据的表达形式很敏感；

由于使用了样本属性独立性的假设，所以如果样本属性有关联时其效果不好。

## 支持向量机

类推主义的主算法是支持向量机。找到合理划分数据的超平面是该算法基本思想。

图示

描述已自动生成

其中法向量ω和常量b通过分别定义超平面的方向与其到原点的距离，使该超平面被唯一确定。此时问题核心为两“支持向量”的间距γ最大化。

若数据不是线性可分的，则需引入核函数将问题简化。该技术采用升维思想，通过非线性映射φ(×)，将原始数据如二维空间转化成更高维的三维空间，即可在高维特征空间中使数据线性可分，并按相似步骤求解。值得注意的是，核函数的函数选取对能否接近问题的最优解有很大影响。依据经验首先试用高斯核函数。

### 支持向量机模型

#### 线性可分SVM

当训练数据线性可分时，通过硬间隔最大化可以学习得到一个线性分类器，即硬间隔SVM。

一个超平面由法向量W和截距b决定,其方程为, 可以规定法向量指向的一侧为正类,另一侧为负类。

为了找到最大间隔超平面，我们可以先选择分离两类数据的两个平行超平面，使得它们之间的距离尽可能大。在这两个超平面范围内的区域称为“间隔”，最大间隔超平面是位于它们正中间的超平面。

图表, 散点图

描述已自动生成

通过拉格朗日乘子法可以求得最大化间隔的方式。

#### 线性SVM

当训练数据不能线性可分但是可以近似线性可分时，通过软间隔最大化也可以学习到一个线性分类器，即软间隔SVM。

现实中训练数据往往不是严格线性可分的，解决该问题的一个办法是允许SVM在少量样本上出错，即将之前的硬间隔最大化条件放宽一点，这就是软间隔。为了使不满足条件的样本点尽可能少，要新增一个对这些点的惩罚项。最常用的是hinge损失：



#### 非线性SVM

当训练数据线性不可分时，通过使用核技巧和软间隔最大化，可以学习到一个非线性SVM。

核技巧的基本思路分为两步:使用一个变换将原空间的数据映射到新空间(例如更高维甚至无穷维的空间)；然后在新空间里用线性方法从训练数据中学习得到模型。

### 优缺点

优点：

由于SVM是一个凸优化问题，所以求得的解一定是全局最优而不是局部最优。

不仅适用于线性线性问题还适用于非线性问题(用核技巧)。

拥有高维样本空间的数据也能用SVM，这是因为数据集的复杂度只取决于支持向量而不是数据集的维度，这在某种意义上避免了“维数灾难”。

理论基础比较完善(例如神经网络就更像一个黑盒子)。

缺点：

二次规划问题求解将涉及m阶矩阵的计算(m为样本的个数), 因此SVM不适用于超大数据集。

只适用于二分类问题。