

Q0r010

0r010101010 e9iarcto

010rgr⁰1010roior0t0toru010:010i0riCFU70:010r9T01GJ010161010T010.00010r00127010 0601D10100110:01DcatorG40701010T0705010T0torre1010101009rQTQ10100rf0rgr9i0040er

Q01010fa1010701070fotrar010109T010r0101Crolut0r⁰f6101010102010t010(01D 10101019f意 0 0 DI 0T01C10100i010781010101010100ro10f01619101010l0n0r070101019101

**第** **5** **章** **数据的深度分析(数据挖掘、机器学习)**



从明细数据，计算一些聚集统计量 (Aggregation), 生成报表 (Report), 观察同比/ 环比的变化. 以及通过数据透视表 (Pivotal Table) 从不同维度观察数据的汇总信息，只 能算是对数据的简单分析。

要从数据里挖掘和提取隐含的模式和规律，简单分析是无能为力的。我们需要在数据 上运行更加复杂的算法，这些算法主要是数据挖掘和机器学习算法。

**5.1** **机器学习与数据挖掘简介**

从广义上来说，机器学习是一种能够赋予机器学习的能力，让它完成直接编程无法完 成的功能的方法。从实践上来说，机器学习是一种通过利用数据，训练出模型，然后使用 模型进行预测的方法。由此可见，数据对于机器学习的意义。可以说数据是原材料，机器 学习是加工工具，模型是产品。我们可以利用这个产品做预测，指导未来的决策。

机器学习和人工智能紧密关联。人工智能的概念自20世纪50年代提出以来，学术界 和产业界不断研究和探索。人工智能的研究大概经历了几个阶段，从早期的逻辑推理，到 中期的专家系统，到近期的机器学习。从另外一个角度来看，人类的智能主要包括归纳总 结和逻辑演绎，对应到人工智能研究的连接主义(如神经网络)和符号主义(如符号推理 系统)。符号主义的主要思想是应用逻辑推理法则，从公理出发，推演整个理论体系。连 接主义的研究策略，以神经网络为例，则试图在研究和了解人脑工作原理的基础上，创造 出一个人工的神经网络，然后利用数据对该网络进行训练，使其具有强大的预测能力。由 此可以看出，人工智能包含机器学习，人工神经网络是机器学习的一种形式。

数据挖掘可以认为是机器学习算法在数据库上的应用，很多数据挖掘中的算法是机器 学习算法在数据库中的优化。数据挖掘能够形成自己的学术圈，是因为它贡献了独特的算 法，其中最著名的是关联规则分析方法——Apriori 算法。Apriori 算法不是机器学习算法 在数据库上的优化，而是由数据挖掘学术圈的学者创造出来的算法。关联规则的功能，可 以通过下面的典故来了解。

在 一 家超市里有 一 个有趣的现象：尿布和啤酒赫然摆在 一 起出售。这个奇怪的举措却

使尿布和啤酒的销量双双增加了。很多人认为这是一个笑话.但它是发生在美国沃尔玛连 锁超市的真实案例，成为一个典故，不断流传。沃尔玛是著名的零售商，拥有世界上最大 的数据仓库系统。为了能够准确了解顾客在其门店的购买习惯.沃尔玛利用数据挖掘方 法，对各个门店的原始交易数据进行分析和挖掘。对顾客的购物行为进行购物篮的关联分 析，可以知道顾客经常一起购买的商品有哪些。

他们有一个意外的发现，跟尿布一起购买最多的商品竟是啤酒!经过实际调查和分 析，揭示了一个隐藏在“尿布与啤酒”背后的美国人的行为模式。在美国，太太经常叮嘱 丈夫，下班后为小孩买尿布，丈夫们在买尿布后，有30%～40%的人又随手带回了他们喜 欢的啤酒。

机器学习的目的是预测(包括分类和回归)。分类是根据输入数据，判别这些数据隶 属于哪个类别 (Category)。回归则是根据输入数据，计算一个输出值 (Numeric)。 输入 数据一般为一个向量，向量的各个分量也称为特征 (Feature), 输出则是一个类别或者一 个数值。

机器学习的基本过程是利用训练数据(包含输入数据和预期输出的分类或者数值)训 练一个模型 (Model), 利用这个模型，就可以对新的实例数据 (Instance) 进行分类和计 算一个预测值。比如在垃圾邮件分类中，输入向量是邮件里是否包含某个单词或者短语的 由0,1构成的向量，表示某个具体的单词或者短语是否出现，输出则是该邮件是否为垃 圾邮件的类别判定。对图像进行分类(假设图像集的每张图像只有某种动物，整个图像集 对应一个动物集合),图像分辨率为320×240,图像为真彩色，即每个像素有三个原色。 输入向量为320×240×3的向量，输出则是某个动物的分类，比如马、牛、羊、猪、狗、 猫等。又比如，在对时间序列数据进行预测的应用中，输入的是在历史时间序列 (Time Series, 比如股票价格)上计算的一系列指标(比如移动平均值 (Moving Average) 等 ) 构成的一个向量，输出则是未来某个时刻的该时间序列的一个具体数值。

在本章中，我们把常用的机器学习和数据挖掘方法进行统一的介绍。这些算法可以进 行简单的分类，其中的一种分类方法是把机器学习方法分为有监督学习 (Supervised Learning) 、 无 监 督 学 习 (Unsupervised Learning) 和半监督学习 (Semi-Supervised Learning) 。(1) 有监督学习是机器学习的一种类别，训练数据由输入特征 (Feature) 和 预期的输出构成，输出可以是一个连续的值(称为回归分析),或者是一个分类的类别标 签(称为分类)。比如给定20张图片及其标签作为训练集，其中10张图片是小狗的图 片，那么其标签为“狗”,另外10张图片是其他物体的图片，那么其标签为“非狗”, 有监督学习的任务就是训练一个模型，当这个模型再遇到新的图片时，能够根据这个图 片是不是狗的图片，给出“狗”和“非狗”的分类结果。在实际应用中，有监督学习的 具体实例包括，输入数据包含疾病的症状，标签是具体的疾病；输入数据包含各种手写 字符的图片，标签是这些手写图片对应的实际字符等。决策树、支持向量机 (SVM) 、 K 最近邻 (KNN) 等算法，都属于有监督学习。(2)无监督学习与有监督学习的区别 是它没有训练样本，直接对数据进行建模。 K-Means 聚类算法就是典型的无监督学习算 法，它的目的是把相似的对象聚集在一起。聚类算法获得的每个类簇 (Cluster), 需要 用户进行观察和判断，以便了解其实际意义。(3)半监督学习是有监督学习和无监督学 习相结合的一种学习方法。它研究如何利用少量的标注 (Annotated) 样本和大量的未 标注样本进行训练和预测的问题。半监督学习包括半监督分类、半监督回归、半监督聚 类和半监督降维算法。



**5.2** **主流机器学习与数据挖掘方法**

**5.2.1** **决策树**

在机器学习中，决策树是这样一个预测模型，它表示对象属性(比如贷款用户的年 龄、是否有工作、是否有房产、信用评分等)和对象类别(是否批准其贷款申请)之间的 一种映射。决策树中的非叶子节点，表示对象属性的判断条件。其分支表示符合节点条件 的所有对象，树的叶子节点表示对象所属的类别。

文献①给出了一个实例，我们通过该实例来了解决策树的基本原理。表5-1是历史上 某银行授予贷款的客户列表。每个记录表示一个客户，表格的各个列表示客户的一些属性 (包括年龄 (Age)、 是否有工作 (Has Job)、是否有房产 (Own House)、信用评价 (Credit Rating) 等)。其中，最后一列 (Class) 表示是否授予该客户贷款申请，也就是客 户的贷款申请是否获得批准。该表格记录了该银行根据不同用户的情况，是否批准其贷款 申请的历史信息。

**表5-1** **客户贷款情况表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Age | Has Job | Own House | Credit Rating | Class |
| 1 | young | false | false | fair | No |
| 2 | young | false | false | good | No |
| 3 | young | true | false | good | Yes |
| 4 | young | true | true | fair | Yes |
| 5 | young | false | false | fair | No |
| 6 | middle | false | false | fair | No |
| 7 | middle | false | false | good | No |
| 8 | middle | true | true | good | Yes |
| 9 | middle | false | true | excellent | Yes |
| 10 | middle | false | true | excellent | Yes |
| 11 | old | false | true | excellent | Yes |
| 12 | old | false | true | good | Yes |
| 13 | old | true | false | good | Yes |
| 14 | old | true | false | excellent | Yes |
| 15 | old | false | false | fair | No |

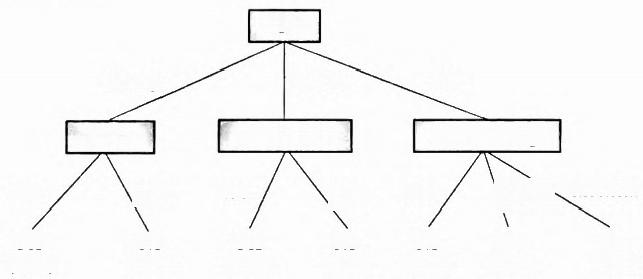
图5-1是从上述历史数据中训练出来的一个决策树。利用该决策树，银行就可以根 据新来客户的一些基本属性，决定是否批准其贷款申请。比如某个新客户的年龄是中年， 拥有房产，那么我们根据其基本信息，沿着决策树的树根一直到叶子节点，得出 “Yes” 的决策结果，即可以批准其贷款申请。具体是，我们首先访问根节点Age, 根据该用户的 年龄为中年，我们应该走中间那个分支，到达是否拥有房产的节点“Own House?”,由 于



①<https://link.springer.com/chapter/10.1007/978-3-540-37882-2_3.>



该客户拥有房产，所以我们走左边那个分支，到达叶子节点，节点的标签是 “Yes”, 也就 是应批准其贷款申请。



Age?

middle old

Own House? Credit Rating?

false good

Yes No No Yes

(3/3) (2/2) (1/1) (2/2)

true

Yes

(2/2)

false

No

(3/3)

true fair

Yes

(2/2)

Has Job?

excellent

young

图5 - 1 从客户贷款情况表训练出来的决策树

决策树可以转化为一系列的规则 (Rule), 从而构成一个规则集 (Rule Set),这样的 规则很容易理解和运用。比如上述决策树，最左边的分支对应的规则是：如果客户年龄属 于青年，而且有工作，那么就可以批准其贷款申请。

1. 决策树的构造过程

决策树的创建从根节点开始，也就是需要确定一个属性，根据不同记录在该属性上的 取值，对所有记录进行划分。接下来，对每个分支重复这个过程，即对每个分支选择另外 一个未参与树的创建的属性，继续对样本进行划分， 一直到某个分支上的样本都属于同一 类(或者隶属该路径的样本大部分属于同一类)。比如在上述实例中，经过树中的一系列 非叶子节点的划分后，样本被分成批准贷款 (Yes) 和未批准贷款 (No) 两类，这样的节 点形成叶子节点。

属性的选择也称为特征选择。特征选择的目的是使分类后的数据集比较纯，即数据 (子)集里主要是某个类别的样本，因为决策树的目标就是把数据集按对应的类别标签进 行分类。理想的情况是，通过特征的选择，能把不同类别的数据集贴上对应的类别标签。 为了衡量一个数据集的纯度，就需要引入数据纯度函数。

其中一个应用广泛的度量函数是信息增益 (Information Gain)。信息熵表示的是不确 定性。非均匀分布时，不确定性最大，此时熵就最大。当选择某个特征，对数据集进行分 类时，分类后的数据集的信息熵会比分类前的小，其差值表示为信息增益。信息增益可以 衡量某个特征对分类结果的影响大小。

对于一个数据集，特征 A 作用之前的信息熵计算公式为：  log₂ (P)。式中，D 为训练数据集； c 为类别数量； P; 为类别i 样本数量占所有样本的 比例。

对应数据集D, 选择特征 A 作为决策树判断节点时，在特征 A 作用后的信息熵为

InfoA(D)(特征A 作用后的信息熵计算公式),计算如下： 式中，k 为样本D 被分为k 个子集。

信息增益表示数据集D 在特征A 的作用后，其信息熵减少的值(信息熵差值),其计 算公式如下：Gain(A)=Info(D)—InfoA(D)。

在决策树的构建过程中，需要选择特征值时，都选择Gain(A) 值最大的特征。

**2.** **决策树的剪枝**

在决策树建立的过程中，很容易出现过拟合 (Overfitting) 的现象。过拟合是指模型 非常逼近训练样本，模型是在训练样本上训练出来的，在训练样本上预测的准确率很高， 但是对测试样本的预测准确率不高，效果并不好，也就是模型的泛化能力 (Generaliza- tion) 差。当把模型应用到新数据上时，其预测效果不好，过拟合不利于模型的实际应用。

决策树同样出现过拟合现象，可以通过剪枝进行一定的修复。剪枝分为预先剪枝和后 剪枝两种情况。预先剪枝指的是在决策树构造过程中，使用一定条件加以限制，在产生完 全拟合的决策树之前就停止生长。预先剪枝的判断方法也有很多，比如信息增益小于一定 阈值时，通过剪枝使决策树停止生长。

后剪枝是在决策树构造完成之后，也就是所有的训练样本都可以用决策树划分到不同 子类以后，按照自底向上的方向，修剪决策树。后剪枝有两种方式：一种是用新的叶子节 点替换子树，该节点的预测类由子树数据集中的多数类决定；另一种是用子树中最常使用 的分支代替子树。后剪枝一般能够产生更好的效果，因为预先剪枝可能过早地终止决策树 构造过程。需要注意的是，后剪枝在子树被剪掉后，决策树构造的一部分计算就浪费了。

决策树算法有一些变种，包括ID3,C4.5,CART 等，一般都需要经过两个阶段来进 行构造，即树的生长阶段 (Growing) 和剪枝阶段 (Pruning)。

决策树的应用非常广泛，除了上述是否批准贷款申请的实例，还可以应用在对客户进 行细分、对垃圾邮件进行识别等场合。

**5.2.2 聚类算法K-Means**

K-Means 算法是最简单的一种聚类算法，属于无监督学习算法。

假设我们的样本是{x¹),x²), … ,x(m)}, 每个x(i)∈R”, 即它是一个n 维向量。现在用 户给定一个 K 值，要求将样本聚类 (Clustering) 成 K 个类簇 (Cluster) 。 在这里，我们 把整个算法称为聚类算法，聚类算法的结果是一系列的类簇。

K-Means 是一个迭代型的算法，它的算法流程是：

|  |
| --- |
| 1. 随机选取 K 个聚类质心(Cluster Centroid), 为μ1,μ2 ·…,μ∈ R”。  2. 重复下面过程，直到收敛：  2.1. 对于每个样本i,计算它应该属于的类：  c(i):=arg min;||x① 一 μ; II²  2.2. 对于每一个类别j,重新计算它的质心：    } |

收敛是在上一次迭代到本次迭代中，每个样本隶属于同样的类别，每个类别的质心不 再发生改变。

下面以一个实例展示K-Mean 标准算法的执行过程。假设我们对样本进行K=3 的聚类， 图5-2中的正方形是样本点，圆形(不同灰度)分别是三个类的质心。

图 5 - 2 (a) 表示，由于K=3, 所以算法开始在有效的数据域 (Domain) 里生成3个初 始的 (Initial) 质心。图5-2 (b) 表示，所有的样本根据和质心的远近，分配到最近的质 心，形成三个类别。图中的三个区域是基于三个质心的对平面的一个Voronoi Diagram划分， 该划分把平面上的点根据到三个质心的远近分配给各个划分。图5-2 (c) 表示，经过迭代 计算，K 个类簇的质心改变了。图5-2(d) 表示经过多次迭代(把样本分配到各个质心、 重新计算各个类簇的质心),各个类簇最终的质心，即聚类算法收敛的结果。

在 K-Means 算法中，涉及距离的计算，最常用的距离是欧式距离。欧式距离 (Eu-

clidean Distance) 的公式为： 。此外，还有闵可夫斯基距离、曼哈 顿距离(也称为城市街区距离， City Block Distance) 可以使用，它们的计算公式分别为：



|  |  |  |
| --- | --- | --- |
| □ □  □  (a) | | (b) |
| (c) |  | □ |
| (d) |

图5-2 K-Means 算法实例

注：图中不同灰度的圆圈不是真实存在的数据点，而是某个类簇的虚拟质心。

在K-Means 算法中，K 值的选择是一个重要的问题。我们希望所选择的K 正好是数 据里隐含的真实的类簇的数目。我们可以选择一个合适的类簇指标，当我们假设的类簇的 数目等于或大于真实的类簇的数目时，这个指标变化平缓，当我们假设的类簇的数目小于 真实的类簇的数目时，这个指标急剧变化。

可以选择的类簇指标包括平均半径或者直径。类簇的直径是指类簇中任意两点的距离 的最大值。类簇的半径是指类簇中所有点到类簇中心距离的最大值。我们可以给出一系列 的 K 值，运行K-Means 算法计算上述类簇指标，然后根据上述原则，选取一个合适的 K 值。

K-Means 是可伸缩和高效的，方便处理大数据集，计算的复杂度为O(NKt), 其 中 N 为数据对象的数目，t 为迭代的次数。 一般来说，K<<N,t<<N。 当各个类簇是密 集的，且类簇与类簇之间区别明显时，K-Means 算法可以取得较好的效果。

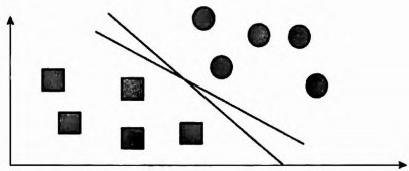
K-Means 算法有三个缺点：(1) K-Means 算法中的K 是事先给定的， 一个合适的K 值难以估计。(2)在K-Means 算法中，首先需要根据初始类簇中心来确定一个初始划分， 然后对初始划分进行优化。初始类簇中心的选择对聚类结果有较大的影响。 一旦初始值选 择的不好，可能无法得到有效的聚类结果。可以使用遗传算法 (Genetic Algorithm), 帮 助选择合适的初始类簇中心。(3)算法需要不断地进行样本分类调整，不断地计算调整后 的新的类簇中心，因此当数据量非常大时，算法的时间开销是非常大的。可以利用采样策 略，改进算法效率。也就是初始点的选择，以及每一次迭代完成时对数据的调整，都是建 立在随机采样的样本数据的基础之上，这样可以提高算法的收敛速度。

**5.2.3 分类算法支持向量机** **(SVM)**

支持向量机由 Boser,Guyon,Vapnik 等在1992年的ACM Conference on Computa- tional Learning Theory(COLT) 会议上提出。从那以后，支持向量机以其在解决小样本、 非线性以及高维模式识别中表现出来的特有优势得到广泛的应用。其应用领域包括文本分 类、图像识别等。

**1.** **二维空间(即平面)数据点的分类**

下面通过一个二维平面上的简单分类来介绍支持向量机分类技术(见图5-3)。在平 面上有两种不同的点，用不同的形状表示，一种为圆形，一种为正方形。现在我们在平面 上绘制一根直线，把两类点分开。这样的直线可以画很多条，那么哪一条才是最好的分割 线呢?



**图5-3** **二维平面上的数据分类(有无数个分割直线)**

在二维平面上，把两类数据分开(假设可以分成两类)需要一条直线。到了3维空 间，要把两类数据分开，就需要一个平面。把上述分类机制扩展到基本情形，在高维空间 里，把两类数据分开，则需要一个超平面。直线和平面是超平面在2维和3维空间的表现 形式。

**2.** **支持向量**

我们寻找分类函数y=f(x)=wTx+b, 超平面上的点代入这个分类函数，得到f(x)=0;

超平面一边的数据点代入分类函数，得到f(x)=1; 超平面另外一边的数据点代入分类函数， 得到f(x)=-1 。 在二维平面上，这个分类函数对应一根直线y=f(x)=ar+b。

在二维平面上确定一根直线，就是确定上述方程中的a 和b 。在高维空间上确定一个 超平面，则需要确定w 向量和b 向量。

如何确定w 和b 呢?答案是寻找一个超平面，它到两个类别数据点的距离都尽可能



大，这样的超平面为最优的超平面。

在图5-4中，中间的那根直线到两类数据点的距离是相等的(图中双向箭头的长度 表示距离d) 。 为了确定这根直线，不需要所有的数据点(向量),只需要图中显示为深灰 色的数据点(向量),这些向量唯一确定了数据划分的直线(超平面),称为支持向量 (Support Vector)。

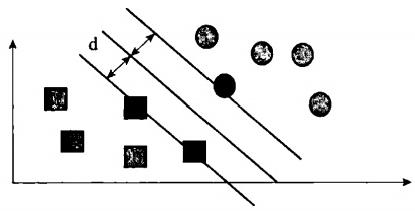


图5-4 支持向量 (d 表示超平面到不同类数据点的距离)

通过上面实例的分析和解释，我们总结如下，支持向量机 (Support Vector Machine, SVM) 是一个对高维数据进行分类的分类器。数据点被划分到两个不相交的半空间 (Half Space), 从而实现分类，划分两个半空间的是一个超平面，SVM 分类的主要任务是寻找 到和两类数据点都具有最大距离的超平面，目的是使得把两类数据点分开的划分范围 (Margin) 最大化。

SVM 问题模型 wT·x++b=+1

*wT·x+b=-1*

wT · (x+-x⁻)=2, 将线性问题转化为求max(Margin Width)



可以看成由样本构成的向量(x+-x⁻), 在分类超平面上的法向量上 的 投影。

假设训练数据为

(x₁,y₁),(x₂,y₂),…,(xn,yn)∈Rd,y∈{+1,-1} 线性分类函数

w·x+b=0,w∈R,b∈R

max-margin问题转化成优化问题



线性SVM问题建模

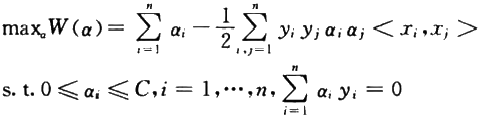
最优分类平面求解问题，表示成约束优化问题

最小化目标函数： 约束条件：y;((w⁷·x;)+b)≥1,i=1,…,n



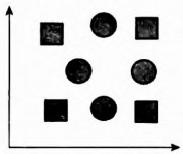
对间隔最大的超平面进行计算是一个二次规划问题。可以通过应用拉格朗日对偶性

(Lagrange Duality)求解对偶问题得到最优解。具体的问题转化和求解过程请参考相关资 料。对偶问题的具体形式如下：



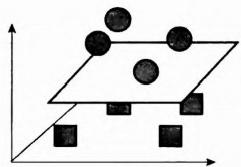
要解决的问题是在参数{a,a₂,…,an} 上求最大值W 的问题，而xi,y; 都是已知数。 在优化模型中的控制参数C 的作用是控制目标函数中的两个目标之间的权重。这两个目标 是寻找间隔最大的超平面和保证数据点分类误差最小。

有时候两个数据点集，在低维空间中，我们无法找到一个超平面来清晰地划分，比如 图 5 - 5 (a) 中的二维平面上的两类数据点，找不到一根直线把它们划分开。 SVM 数据分 析方法里有一个核函数 (Kernel Function) 技巧，可以巧妙地解决这个问题。通过核函 数，可以把低维空间的数据点(向量)映射到高维空间中。经过映射以后，两类数据点在 高维空间里，可以有一个超平面把它们分开。



(a) 二维平面上的两类数据点

(不易分类)



(b) 经过核函数映射后的三维空间上的两类数据点

(容易分类)

图5-5 SVM的核函数技巧

在 图 5 - 5 (b) 中，两类数据点经过映射以后，映射到3维空间的数据点。在3维空 间里，我们可以看到，两类数据点可以用一个平面分开。需要注意的是，图5-5仅仅通 过举例说明低维线性不可划分的两类向量，可以通过核函数映射到高维空间，从而达到线 性可划分的目的。图5-5的目的是说明核函数的作用，是把低维空间线性不可分的向量， 映射到高维空间，使其线性可分。

常用的核函数包括多项式核函数、高斯径向基核函数、指数径向基核函数、多隐层感 知核函数、傅立叶级数核函数、样条核函数、 B 样条核函数等。

3. 异常值的处理

在上述SVM 模型中，超平面本身是由少数几个支持向量 (Support Vector) 确定的， 未考虑Outlier (离群值，异常值)的影响。我们通过下面的实例，展示异常值的影响。

在图5-6里，不同形状表示不同类别的数据点，深灰色的数据点则表示异常值。这 个异常值将导致分类错误。我们的目标是计算一个分割超平面，把分类错误降到最小。

这个问题通过在模型中引入松弛变量 (Relax Variable) 来解决。松弛变量是为了纠 正或约束少量“不安分”或脱离集体不好归类的数据点的因子。引入松弛变量后，支持向 量机的超平面的求解问题仍然可以转化成一个二次规划问题求解。在引入松弛变量的情况 下，最优分类平面求解问题表示成约束优化问题。



最小化目标函数：

约束条件：y;((w·x;)+b)≥1-ξ,≥0,i=1,2,…,n

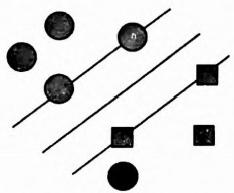


图5-6 SVM 分类器中异常值的处理

正则化参数C, 可以理解为允许划分错误的权重，C 越大，表示越不允许出错，当C 较小 时，允许少量样本划分错误。至此，我们看到，支持向量机不仅能够处理线性分类，还能够处 理数据中的非线性(使用核函数)、容忍异常值(使用松弛变量),是一个强大的分类器。

4. 支持向量回归

支持向量机的训练数据为{(x₁,y₁),(x₂,y₂),…,(xn,y)},Ii∈R”,y;∈{-1,

+1}。如果训练数据中的y, 为一个实数，即y;∈R, 则引出了支持向量回归 (Support Vector Regression,SVR) 问题。

给定训练数据集S, 以及任意给定ε>0。如果在原始空间R” 空间存在一个超平面 f(x)=<w,x >+b,w∈R”,b∈R,使 得 |y,—f(x;)|<=e,V(x;,y;)∈S, 则称 f(x)=<w,x>+b 是样本集合S 的ε-线性回归。问题转化为S 中任何点(x;,y;) 到超平

面f(x)=<w,x>+b 的距离不超过 ,也就是最大化 ,等价

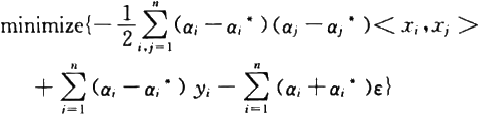
于min{||w |I²}。

∈-线性回归问题转化为优化问题：



s.t.|(w⁷·x;)+b-y;l≤e,i=1,2,…,n

和解决SVM 分类问题一样，可以引入松弛变量。利用Lagrange 对偶性，得到优化问 题的对偶性形式，再进行求解。





对于不可能在原始空间R” 可以线性回归的训练数据集S, 首先用一个非线性函数 φ(S) 将数据集S 映射到一个高维空间，在高维空间φ(S) 具有很好的线性回归特性，然后 在高维空间进行ε一线性回归。

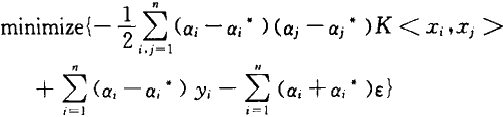
于是，支持向量回归分析的过程如下：



已知训练集{(x₁,y₁),(x₂,y₂),…,(xn,yn)), 其中xi∈R”,y∈R,i=1,2,…,n, 选 择适当的正数ε和C。

(1)寻找一个核函数 K(s,t), 使得 K(x;,xj)=<φ(x;),φ(x;)>。

(2)求优化问题的解αi,ai"。





得到最优解α1,α1”,α2,α2,…,an,an"。

(3)计算参数b, 选择开区间(0,C) 中的α;或者 αk, 如果选择到的是α,,那么



如果选择到的是αk\*, 那么



(4)最后构造非线性函数。



5.SVR 的特点

可以把非线性函数f(x) 看作2维空间的一条曲线(多维空间的一个曲面),训练数 据点分布在这个曲线周围，形成一个“管道”。在所有训练数据点中，只有分布在“管道” 壁上的那些训练数据点，才决定管道的位置。于是这些训练数据点就称为支持向量。

具体到2维平面空间， SVM 的目的是通过一条直线，尽量把两类数据点分开。 SVR 的目的则是通过一个管道，尽量把数据点都囊括进来。

为了适应训练数据集的非线性 (Non-Linearity), 传统的拟合方法通常在线性方程里

加上高阶项来解决。这种方法一般是有效的，但是也增加了过拟合的风险。

支持向量回归算法采用核函数解决非线性问题。引进核函数的目的是“升维”,在低 维空间中的非线性转换成高维空间的线性问题，把非线性回归转换成线性回归，然后求 解。或者反过来说，用核函数代替线性方程中的线性项，使得线性回归算法非线性化，即 能做非线性回归。SVR 的“升维”虽然增加了可调参数，但是很好地控制了过拟合问题。

5.2.4 关联规则分析 Apriori 算 法

关联规则分析 (Association Rule Analysis) 最典型的例子是购物篮分析。通过关联规 则分析，能够发现顾客每次购物的购物篮中不同商品之间的关联，从而了解客户的消费习 惯，让商家能够了解哪些商品被客户同时购买，帮助他们制定更好的营销方案。本章开头



**提到的“尿布与啤酒”的典故，就是关联规则分析方法的挖掘结果**。

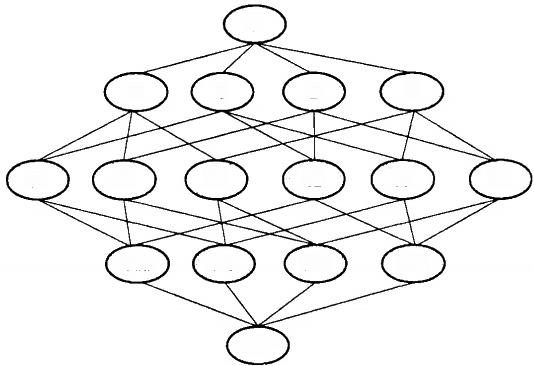
**关联规则是形如X→Y 的蕴含式.表示通过X 可以推导出Y,X 称为关联规则的左部** **(Left Hand Side,LHS),Y 称为关联规则的右部** **(Right Hand Side,RHS)。在购物篮** **分析结果里，尿布→啤酒表示客户在购买尿布的同时，有很大的可能性购买啤酒。**

关联规则有两个指标，分别是支持度 (Support) 和置信度 (Confidence) 。 关联规则 A→B的支持度 (support)=P(AB), 指的是事件 A 和事件 B 同时发生的概率。置信度 (confidence)=P(B|A)=P(AB)/P(A), 指的是发生事件A 的基础上，发生事件B 的 概率。比如，如果尿布 → 啤酒关联规则的支持度为30%,置信度为60%,那么就表示所 有的商品交易中，30%交易同时购买了尿布和啤酒，在购买尿布的交易中，60%的交易同 时购买了啤酒。

关联规则分析需要从基础数据中挖掘出支持度和置信度都超过一定阈值的关联规则， 以便在决策中应用。同时满足最小支持度阈值和最小置信度阈值的规则，称为强规则。

挖掘关联规则的主流算法为 Apriori 算法。它的基本原理是在数据集中找出同时出现 概率符合预定义 (Pre-defined) 支持度的频繁项集.而后从以上频繁项集中，找出符合预 定义置信度的关联规则。频繁项集和关联规则可以通过以下实例来解释。

假设有一家商店经营4种商品(实际生活中，商品数目比这大得多.但是不影响算法 原理的阐述),分别是商品0、商品1、商品2和商品3。那么所有商品的组合有：只包含 一种商品的、包含两种商品的、包含三种商品的以及包含四种商品的组合。这些组合，包 括空集，构成如图5-7所示的子集或者超集关系。图5-7中的圆圈表示某个商品组合， 连接线则表示子集/超集关系。



φ

1 2

03 12

013 023

0123

012

123

23

01

02

13

0

3

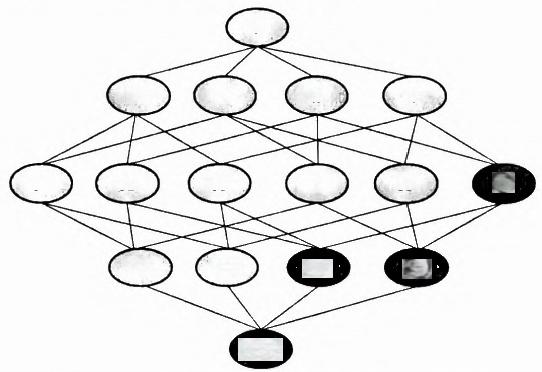
**图5-7** **集合〈商品0,商品1,商品2,商品3}中所有可能的项集组合**

对于单个项集的支持度，我们可以通过遍历每条记录并检查该记录是否包含该项集来 计算。对于包含 N 种物品的数据集共有2-1种项集组合，重复上述计算过程是不现实 的。科研人员发现Apriori 原理可以减少计算量。

Apriori 原理是，如果某个项集是频繁的，那么它的所有子集也是频繁的。它的逆否 命题是，如果一个项集是非频繁的，那么它的所有超集也是非频繁的。比如在图5-8中， 已知阴影项集{商品2,商品3}是非频繁的。利用这个基础知识，我们可以知道项集 {商品0,商品2,商品3},〈商品1 . 商品2,商品3}以及{商品0,商品1,商品2,商

品3}也是非频繁的，因为它们都是{商品2,商品3}的超集。

于是在计算过程中， 一旦计算出(商品2,商品3}的支持度，知道它是非频繁的后， 就可以紧接着排除{商品0,商品2,商品3},{商品1,商品2,商品3}和〈商品0,商 品1,商品2,商品3}项集的判断，于是节省了计算工作量。



中

3

13

123

**0123**

023

23

012

013

01

03

02

12

2

0

1

**图5-8** **Apriori 原理(非频繁项集用深灰色表示)**

除 了Apriori 算法，挖掘关联规则的算法还有FP-growth 。FP-growth 算法基于Apri- ori 算法构建，它采用优化的数据结构，减少扫描次数。 FP-growth 算法只需对数据库进 行两次扫描，大大加快了算法执行速度。Apriori算法对于每个潜在的频繁项集，都会扫 描数据集判定给定模式是否频繁。

文献①给出了一个实例，我们通过该实例来了解 Apriori 算法的执行过程。假设有如 下的数据集，表示用户观看电影的活动。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 交易号 | 电影1 | 电影2 | 电影3 | 电影4 | 电影5 |
| 1 | Sixth Sense | LOTR1 | Harry Potterl | Green Mile | LOTR2 |
| 2 | Gladiator | Patriot | Braveheart |  |  |
| 3 | LOTR1 | LOTR2 |  |  |  |
| 4 | Gladiator | Patriot | Sixth Sense |  |  |
| 5 | Gladiator | Patriot | Sixth Sense |  |  |
| 6 | Gladiator | Patriot | Sixth Sense |  |  |
| 7 | Harry Potterl | Harry Potter2 |  |  |  |
| 8 | Gladiator | Patriot |  |  |  |
| 9 | Gladiator | Patriot | Sixth Sense |  |  |
| 10 | Sixth Sense | LOTR | Gladiator | Green Mile |  |

我们的目标是从这些数据中挖掘出用户观看电影的关联规则，购票观看某部(几部) 电影的用户中，有多少比例会去购票观看其他哪些电影。支持度阈值设定为50%,置信度 阈值设定为80%。



①<http://it.taocms.org/10/1833.htm.>



根据上述数据，我们得到1项集，列表如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 电影名称 | 同时出现在哪些交易中 | 出现次数 | 支持度 |
| Sixth Sense | 1,4,5,6,9,10 | 6 | 6/10=0.6 |
| LOTR1 | 1,3 | 2 | 2/10=0.2 |
| Harry Potterl | 1 | 1 | 1/10=0.1 |
| Green Mile | 1,10 | 2 | 2/10=0.2 |
| LOTR2 | 1,3 | 2 | 2/10=0.2 |
| Gladiator | 2,4,5,6,8,9,10 | 7 | 7/10=0.7 |
| Patriot | 2,4,5,6,8,9 | 6 | 6/10=0.6 |
| Braveheart | 2 | 1 | 1/10=0.1 |
| Harry Potter2 | 7 | 1 | 1/10=0.1 |
| LOTR | 10 | 1 | 1/10=0.1 |

从这张表中可以看出，符合支持度≥50%的只有如下三项，即有3个1项集是频 繁 的 。

|  |  |  |  |
| --- | --- | --- | --- |
| 电影名称 | 同时出现在哪些交易中 | 出现次数 | 支持度 |
| Sixth Sense | 1,4,5,6,9,10 | 6 | 6/10=0.6 |
| Gladiator | 2,4,5,6,8,9,10 | 7 | 7/10=0.7 |
| Patriot | 2,4,5,6,8,9 | 6 | 6/10=0.6 |

在此基础上，通过自连接构造超集，构造出来的2项集为：

|  |  |  |  |
| --- | --- | --- | --- |
| 电影名称 | 同时出现在哪些交易中 | 出现次数 | 支持度 |
| Sixth Sense,Gladiator | 4,5,6,9,10 | 5 | 6/10=0.6 |
| Sixth Sense,Patriot | 4,5,6,9 | 4 | 4/10=0.4 |
| Gladiator.Patriot | 2.4,5,6,8,9 | 6 | 6/10=0.6 |

可以看出，符合支持度≥50%的只有如下两项，即有2个2项集是频繁的。

|  |  |  |  |
| --- | --- | --- | --- |
| 电影名称 | 同时出现在哪些交易中 | 出现次数 | 支持度 |
| Sixth Sense,Gladiator | 4,5,6,9,10 | 5 | 5/10=0.5 |
| Gladiator.Patriot | 2,4,5,6.8,9 | 6 | 6/10=0.6 |

在此基础上，通过自连接构造超集，构造出来的3项集如下，支持度为40%,是不频繁的。

|  |  |  |  |
| --- | --- | --- | --- |
| 电影名称 | 同时出现在哪些交易中 | 出现次数 | 支持度 |
| Sixth Sense,Gladiator Patriot | 4,5.6,9 | 4 | 4/10=0.4 |

Apriori 算法利用了上文讲述的 Apriori 原理。上述实例中，因为2项集{Sixth Sense,Patriot} 是不频繁的(基于一定的支持度),包含{Sixth Sense,Patriot} 的 3 项 集，必定是不频繁的，无须继续判断。

到这里，我们提取了如下关联规则。



|  |  |  |
| --- | --- | --- |
| 规则 | 支持度 | 置信度 |
| Sixth Sense=>Gladiator | 0.5 | 5/6=0.8333333 |
| Gladiator=>Sixth Sense | 0.5 | 5/7=0.7142857 |
| Gladiator=>Patriot | 0.6 | 6/7=0.8571429 |
| Patriot=>Gladiator | 0.6 | 1.0 |

由于置信度的阈值为80%,所以第二个关联规则不符合要求。最终提取的关联规则如下：

|  |  |  |
| --- | --- | --- |
| 规则 | 支持度 | 置信度 |
| Sixth Sense=>Gladiator | 0.5 | 5/6=0.8333333 |
| Gladiator=>Patriot | 0.6 | 6/7=0.8571429 |
| Patriot=>Gladiator | 0.6 | 1.0 |

对于关联规则，我们可以很容易地解释和理解。比如，对于上述实例，可以给出如 下解释：(1)购票观看 Sixth Sense(《第六感》)的用户中，有83%会购票观看Gladiator (《角斗士》),同时购票观看这两部电影的用户占50%。(2)购票观看 Gladiator的用户 中，有85.7%会购票观看 Patriot(《 爱国者》),同时购票观看这两部电影的用户占 60%。(3)购票观看Patriot 的用户中，100%会购票观看Gladiator, 同时购票观看这两 部电影的用户占60%。

**5.2.5 EM 算** **法**

EM(Expectation Maximization, 最大期望)算法是为概率分布模型 (Probability Distribution Model) 寻找参数的最大似然估计的算法。这个概率模型包含无法观测的隐藏 变量 (Latent Variable) 。EM算法用于机器学习的聚类 (Clustering) 。 下面我们通过实 例①来了解最大似然估计以及EM 算法。

**1.** **最大似然估计** **(Maximum** **Likelihood** **Estimation)**

假设我们用硬币 A 和硬币B 进行投掷硬币实验。这两个硬币在进行投掷实验时.正 面朝上的概率分别为 θA 和 θB 。我们进行了5轮投掷实验，每轮实验要么选择硬币A, 要 么选择硬币B. 一共连续投掷10次，每次投掷结果要么正面朝上(用H 表示).要么背面 朝 上 ( 用T 表示)。于是我们得到5个正面/背面标志列表。

在实验过程中，我们记录了两个向量列表 x=(x₁,x₂·…·x₅) 以 及 z=(z1 ·

z2 ·… ·z₅). 其中x;∈{0.1.….10}. 记录第i轮投掷实验中 ·正面朝上的次数。z∈

{A,B}, 则是第i 轮投掷实验使用的硬币。假设实验的结果如表5-2所示。

**表5-2** **硬币投掷实验结果**

|  |  |  |
| --- | --- | --- |
| i | x | z |
| 1 | HTTTH HTHTH | B |
| 2 | HHHHT HHHHH | A |
| 3 | HTHHH HHTHH | A |
| 4 | HTHTT THHTT | B |
| 5 | THHHT HHHTH | A |

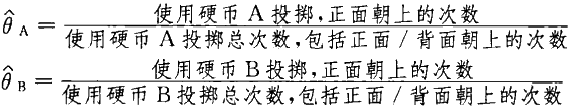


①<http://ai.stanford.edu/>～chuongdo/papers/em\_tutorial.pdf.



目前，我们不知道 θA 和 θB的具体取值，我们只有上述实验结果。在这里我们对 θA 和 θB 做出估计，我们依赖于完整的数据(拥有所有相关的随机变量的取值),包括每轮投掷 用硬币A 还是硬币B, 每轮投掷的10次投掷的正面/背面情况。

那么对 θA 和 s 进行估计的简单方法是，根据实验数据，计算硬币A 和硬币B 的正 面朝上的比例。使用如下公式进行计算：



针对上述实例，我们的计算结果如表5-3所示。

**表5-3** **最大似然估计**

|  |  |  |
| --- | --- | --- |
| 硬币A | 硬币B | 参数估计 |
|  | 5H,5T |  |
| 9H,1T |  |
| 8H,2T |  |
|  | 4H,6T |
| 7H.3T |  |
| 合计24H,6T | 合计9H.11T |

上述方法即最大似然估计法。最大似然估计是参数估计的方法之一。已知某个随机样 本满足某种概率分布，但是其中具体的参数不清楚，参数估计就是通过若干次试验，观察 其结果，利用结果推出参数的大概值。最大似然估计的核心思想是，如果某个参数能使这 个样本出现的概率最大，我们干脆就把这个参数作为估计的真实值。假设logP(x,z;θ) 是 x 和z 的联合分布的对数 (Logarithm), 对其最大化问题进行求解，那么上述对 θA 和 θB 进行估计的公式就是其解。

2.EM 算法实例

现在我们考虑参数估计的另外一种情形。我们仅仅获得了向量列表x=(x₁,x₂, … ,

x₅), 但是没有对应的z=(z1,z2,…,z₅)。 也就是我们知道每轮投掷的10次投掷的正面/ 背面情况.但不知道每轮是用硬币A 还是硬币 B。

我们把z 称为隐藏变量 (Hidden Variable) 或者潜在因子 (Latent Factor)。在这种

情况下，参数估计是在数据不全 (Imcomplete Data) 的情况下进行的。

为了对 θA 和 θs 做出估计，需要知道每轮正面/背面序列是使用硬币A 还是硬币 B。 要知道每轮正面/背面序列是使用硬币A 还是硬币B, 则需要对ôA 和 B 有一个合理的 估计，两者互为前提。事情好像陷入死锁了，必须打破这个循环依赖，才能开始参数的估 计，这就需要用到EM 算法。

EM 算法在两个步骤，即在Expectation 步和 Maximization 步之间进行交替，经过若 干次迭代后，对模型的参数给出合理的估计。

对于这个实例，具体的过程如下：

(1)首先，给出 θA 和 θB 两个参数的初始估计，比如 A=0.6, B=0.5。

( 2 ) 在Expectation 步，使用当前参数，计算每轮投掷的硬币是A 的概率/是B 的概 率，然后计算每轮投掷的正面朝上/背面朝上的次数。



需要使用贝叶斯定理





比如根据第一轮的投掷结果，我们不知道是硬币A 还是硬币B, 给这两个硬币的概率 设定为0.5,即P(A)=P(B)=0.5 。 然后根据第一轮投掷结果5个正面朝上，5个背面

朝上，那么P(x/A)=0.6⁵×0.45,P(x/B)=0.5⁵×0.55 (注意 θA=0.6,θB=0.5), P(A)=0.5,P(B)=0.5, 代入上述等式，得到 P(A|x,θ)=0.45,P(B|r,θ)=

0.55(即图5-9中第二步的第一行数据)。

( 3 ) 在Maximization 步，利用Expectation 获得每轮投掷的硬币是A 的概率/是B的 概率，然后计算每轮投掷的正面朝上/背面朝上的次数，对 θA 和 θB 两个参数进行估计。 也就是，针对每轮投掷，我们在观测到的x 上，看正面朝上/背面朝上的概率是多少。

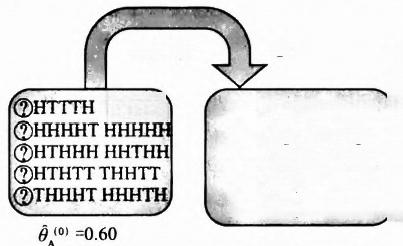
第一轮投掷硬币是A 的概率为0.45,硬币是B 的概率为0.55,按照第一轮投掷5个 正面5个背面的比例，分配到硬币 A 和硬币 B 的正面/背面次数分别是2.2H/2.2T,

2.8H/2.8T。 第2,3,4,5轮的计算类似，请参考图5-9的第三部分的表格。

那么,我们可以汇总出硬币A 正面朝上21.3次，背面朝上8.6次， 0.71,硬币B 正面朝上11.7次，背面朝上8.4次，

(4)经过多次Expectation 步和Maximization 步的循环迭代之后，ôA和 θs 两个参数 收敛。整个过程如图5-9所示。经过10轮Expectation 步和 Maximization 步迭代之后， 6A 和 B 分别为0.8和0.52,和最大似然估计得出的结果已经比较接近了。

|  |  |
| --- | --- |
| **CoinA** | **Coin** **⑧** |
| ≈2.2H,2.2T | ≈2.8H,2.8T |
| ≈7.2 H,0.8T | ≈1.8 H,0.2T |
| ≈5.9H,1.5T | ≈2.1H,0.5T |
| ≈1.4 H,2.1T | ≈2.6H,3.9T |
| ≈4.5 H,1.9T | ≈2.5H,1.1T |
| ≈21.3 H,8.6T | ≈11.7H,8.4T |



E-Step ②

HTHTH 0.45×A,0.55× ⑬

0.80×A,0.20×B

0.73×A,0.27×B

0.35×A,0.65× ⑬

0.65×4,0.35×⑬

g(0)=0.50







θ(10)≈0.80 (10)≈0.52 ④

图5-9 EM算法实例

下面我们再举一个例子，这个例子根据观测数据，估计混合高斯模型的参数。

假设现在有数据Y, 用两个高斯分布φo(x),pa₂(x) 对密度建模，参数为θ=(μ,o²), 那么Y 的概率密度为g(y)=(1—π)φe(y)+πpa₂(y), 其参数为θ=(π,μ1,o₁²,μ2,o2²), π为混合比例。



基于N 个训练数据的对数似然函数是：



直接最大化上述似然函数很难，对其进行变形。引入取值为0或者1的潜在变量△。 如果△;=0,Y, 取自模型1,如果△=1,则Y, 取自模型2。对数似然函数写为：



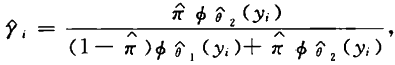


于是(μ1,o₁²) 的极大似然估计是△;=0的那些数据的样本均值和方差，(μ2,o2²) 的极 大似然估计是△;=1的那些数据的样本均值和方差。

由于△的实际值是未知的(也就是隐藏变量),所以用迭代方式进行处理，用下面的 期望代替上述△,也就是γ;(θ)=E(△ |θ.Z)=Pr(△;=1|θ,Z)。

二分量高斯参数估计的 EM 算法如下：

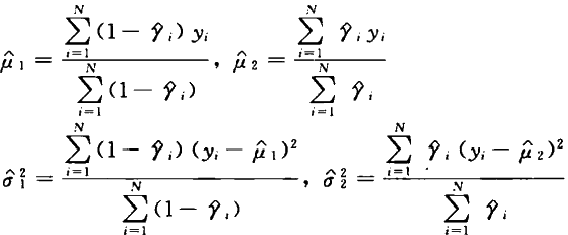
(1)初始化参数1,食2,6²,62,允，其中食i,p2 可以随机选择两个 y:,ô², 经 取

样本的方差 ,混合比例允取0.5。

(2)Expectation 步，上述参数代入，计算 i=1,2,…,N。

表示数据 y: 属于φe 的概率。注意上述计算中，分子为允φ合2(y;), 那是因为9;越 接近0,表示y; 越可能属于模型1,夕；越接近1,则y; 越可能属于模型2。

(3)Maximization 步，计算加权均值和方差，以及混合比例。





表示数据属于φo₂ 的概率总和。

(4)重复步骤(2)和(3)直到收敛。

|  |  |  |
| --- | --- | --- |
| 观测数据y, | 属于模型1(1-夕.) | 属于模型2(9;) |
| yi | 1-91 | 夕1 |
| y2 | 1-92 | 夕2 |
|  | … | \* 非 |
| yN | 1-9N | 9n |

Christopher D.Manning,Prabhakar Raghavan 和 Hinrich Schütze 合著的 An Intro- duction to Information Retrieval, 在16 .5节利用EM 算法，对文档集 (Document Set) 进行基于模型的软聚类 (Soft Clustering)。① 比如，如果有一篇关于中国汽车的文档，聚 类算法把该文档进行软分配 (Soft Assignment),以0.5的概率，隶属于 (Membership) “中国”和“汽车”两个类簇。

Mitchell著的Machine Learning 一书给出了另外一个 EM 的实例。该实例将混搅在一 起的男生和女生身高信息聚成两类。男生身高符合高斯分布，女生身高也符合高斯分布， 但是参数不同。现在男女生的身高混合在一起，问题就变成了，如何估计每个样例是男生 还是女生，然后在确定男女生的情况下，如何估计均值和方差。该书给出了具体的公式， 有兴趣的读者可以进一步参考。

3. 通用EM 算法

给定训练样本集{x(1,x(²2),…,x(n)}, 样本之间独立。这是从参数为θ的分布的总 体样本中，抽取到的若干样本。如果抽取各个样本是相互独立的，那么抽取到这些样本的 概率是样本集 x 中各个样本的联合概率，表达为：



在极大似然估计法中，我们需要找到参数θ,使得似然函数L(θ)最大，称为θ的最大 似然估计θ= argmaxL(θ) 。 由于上述式子中各个概率值是连乘的，不便于分析，可以定 义对数似然函数，变成：



最大似然函数可以通过求导和解方程进行求解。

如果模型中包含隐藏变量(类似于上文提到的投掷硬币的例子，不知道每一轮投掷的 是硬币A 还是硬币B), 每个样本i 对应的类别z(5),z(1 是未知的，也即隐藏变量。

我们对每个样本的每个可能类别z¹ 求联合分布概率。我们知道样本符合某种分布， 但是不知道分布参数θ。因为存在隐藏变量z, 所以直接求θ比较困难，但是一旦确定了z 求解θ就容易了。那么,我们需要找到每个样本隐含的类别z①, 使得 p(x,z) 最大，则 p(x,z) 的最大似然函数为：



对于每个样本 x(, 让 Q 表示该样本隐藏变量 z 的某种分布， Q 满足的条件是

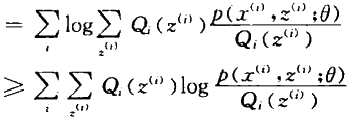
,Q;(z)≥0 。 如果z 是连续的，Q 是概率密度函数。比如，如果要将班上 的学生聚类，假设隐藏变量z 是身高，那么就是连续的高斯分布。如果隐藏变量是男/女， 那么就是伯努利二项分布。

使用Jensen 不等式，对上述似然函数进行变形：

(1)



①<http://nlp.stanford.edu/IR-book/pd>í/irbookprint.pdf.



(2)

(3)

其中，从式(1)到式(2),只需在分子分母同乘以一个相等的函数，从式(2)到式 (3),利用了Jesen 不等式。

基于上述推导过程， EM 算法设计如下。

EM 算法是一种从不完全数据或有数据丢失的数据集(存在隐藏变量)中，求解概率 模型参数的最大似然估计的方法。通用EM 算法的基本流程是：

(1)初始化分布的参数θ。

(2)重复E 步骤和M 步骤，直到收敛。

(2.1)E 步骤，估计隐藏变量。即根据参数初始值或者上一次迭代的模型参数，来计 算出隐藏变量的后验概率，也就是隐藏变量的期望值，作为隐藏变量的当前估计值。Q(z) 是在观察变量已知的情况下z 的分布函数，可以理解为z 的概率密度函数。计算Q(z) 时 ， 观察变量以及参数θ都是已知的。

Q,(z”)=p(z①|x(i;θ)

(2.2)M 步骤，估计其他参数。即根据计算得到的Q. 最大化含有θ的似然函数，获 得新的参数值θ。



(3)通过不断迭代，就可以得到使似然函数 L(θ)最大化的参数θ。本质上讲， EM 的 E 步骤、固定θ,优化Q;M 步骤固定Q, 优化θ。这两个步骤交替，将极值推向最大。

4.EM 算法的应用

EM 算法是对数据进行聚类 (Clustering) 的常用方法。 EM 算法的应用非常广泛，包 括计算机视觉、自然语言处理、心理学、生物信息学等领域，都可以找到它的用武之地。

5.2.6 协 同 过 滤 推 荐 算 法 (Collaborative Filtering Recommendation)

在互联网时代，特别是随着 Web 2.0的发展， Web 已经变成数据分享的平台，每天 都有大量的图片、博客、视频发布到网上。信息的极度爆炸，使得人们找到所需的信息越 来越难。

面对海量的数据，用户需要更加智能的、更加了解他们需求、口味和偏好的信息检索 机制，于是推荐系统应运而生。推荐算法能够根据用户的偏好，向用户推荐商品或者服 务。在电子商务 (E-Commerce, 比 如 Amazon 等)网站、音乐、电影和图书分享网站， 推荐引擎取得了巨大的成功。

推荐引擎需要根据一定的数据进行分析，然后给出推荐结果。主要的数据源包括：

(1)要推荐的物品或者内容的描述信息；(2)用户的基本信息，比如性别、年龄等；

(3)用户对物品或者内容的偏好。用户偏好可以分为两类，分别是显式的用户反馈和隐式 的用户反馈。显式的用户反馈是用户在网站浏览之外，显式地提供的反馈信息，比如用户

**对物品的评分、评论等。隐式的用户反馈则是用户在使用网站时产生的数据，比如用户查** **看了某个物品的信息，用户购买了某个物品等。推荐引擎使用上述数据源，对用户对物品** **的偏好进行预测计算，然后推荐这些物品。**

我们可以从不同的角度对推荐系统进行分类：(1)根据是否为不同用户推荐不同的物 品或者内容，推荐系统分为个性化推荐系统和大众化推荐系统。(2)根据使用的数据源， 推荐系统分为基于内容的推荐系统 (Content-Based Recommendation,它根据推荐物品或 内容的描述，发现物品或者内容的相似性，进行推荐)、基于人口统计学的推荐系统 (De- mographic-Based Recommendation, 它根据用户的信息，发现用户之间的相似性，进行推 荐)、基于协同过滤的推荐系统 (Collaborative Filtering-Based Recommendation,它 根 据 用户对物品或者内容的偏好，发现物品或者内容之间的相似性，或者发现用户之间的相似 性，进行推荐)。(3)根据推荐模型的基本技术原理，推荐系统分为基于用户对物品的评 价矩阵的推荐系统、基于关联规则的推荐系统 (Rule-Based Recommendation, 它通过关 联规则的挖掘，找到哪些物品经常被同时购买，或者用户购买了一些物品后通常会购买哪 些其他的物品，进行推荐)、基于模型的推荐 (Model-Based Recommendation,将已有的 用户偏好信息作为训练样本，训练出一个模型，用于预测用户的其他偏好。以后用户再进 人系统时，可以基于这个模型进行推荐)。基于模型的推荐需要考虑如何利用用户近期或 者实时的偏好信息，更新训练好的模型，以提高推荐的准确度。

**1.** **基于内容的推荐**

基于内容的推荐的基本思路是，根据物品或者内容的描述信息，发现它们之间的相似 性，然后基于用户以往的偏好历史记录，推荐相似的物品或者内容。

比如在电影推荐系统里，我们首先需要对电影的描述信息(元数据)进行建模，这个 模型可以包含电影的类型、电影的导演、电影的演员、电影的剧情介绍等。对于具体用 户，根据他历史上喜欢看的电影，可以给他推荐类似的电影。需要根据上述描述信息，计 算电影之间的相似度。

基于内容的推荐存在若干问题：(1)推荐的质量依赖于物品模型的完整和全面程度；

(2)物品相似度的分析没有考虑人对物品的态度；(3)需要基于用户以往的偏好历史做出 推荐，于是存在“冷启动”(Cold Start) 问题。“冷启动”问题是模型刚开始运行时缺乏 必要的历史数据。

虽然这个方法有这些不足，但是仍然在电影、音乐、图书推荐应用中取得成功。

**2.** **基于人口统计学的推荐**

基于人口统计学的推荐机制是一种最易于实现的推荐。它根据用户的基本信息发现用 户的相似性，然后将相似用户喜爱的其他物品推荐给当前用户。具体来讲，系统对每个用 户都有一个用户画像 (Profile), 这个画像包括用户的一些基本信息，包括年龄、性别、 教育背景、收入、婚姻状况等。然后，系统根据用户的画像，计算用户的相似度。相似用 户称为“近邻”,对于某个用户，利用其“近邻”用户群的偏好，就可以给他一些物品的 推荐。

该方法的优势包括：(1)对于系统的新用户来讲，系统没有“冷启动”问题，因为它 不使用用户对物品的偏好历史数据；(2)该方法不依赖于物品数据，因此它是领域独立的 (Domain Independent), 也就是可以应用到不同领域。但是该方法过于粗糙，对品位要求 较高的领域不适用，比如图书、电影和音乐等领域，推荐效果不是很好。

**3.** **基于协同过滤的推荐**

基于协同过滤的推荐，根据用户对物品或者内容的偏好，发现物品或者内容之间的相 似性，或者发现用户之间的相似性，然后基于这些相似性进行推荐。基于协同过滤的推 荐，又可以分为三个子类：(1)基于用户的 (User-Based) 推荐；(2)基于物品的(项目 的 )(Item-Based) 推荐；(3)基于模型的 (Model-Based) 推荐。

文献①②对基于协同过滤的推荐，进行了详细介绍，这里简述如下。

(1)基于用户的协同过滤推荐。

基于用户的协同过滤推荐，是根据用户对物品或者内容的偏好，发现与某个用户偏好 相似的k 个“近邻”用户(可以使用kNN 算法进行计算),然后基于这k 个“近邻”用户 的历史偏好信息，为该用户进行推荐。以电商领域为例，比如我们现在要给用户C 进行商 品推荐。我们首先进行用户间相似度的计算，发现用户C 和用户 D/E 的相似度较高，也 就是说用户D 和 E 是用户C 的 “k 最近邻”。于是，我们可以给用户C 推荐用户D 和 E 浏 览过或者购买过的商品。需要注意的是，我们只需推荐用户C 还没有浏览过或者购买过的 商品即可，对于用户C 浏览过或者购买过的商品，无须重复推荐。

如果有多个商品可以推荐，到底优先推荐哪个商品呢?需要对商品进行适当的排序。 可以采用加权排序方法，举例如下。比如用户C 的近邻用户为D 和 E, 他们和用户C 的相 似度以及他们评价过的商品的评分如表5-4所示。我们把用户D,E 对其他商品(包括商 品101,102,103)的评分，乘以用户D,E 和用户C 之间的相似度，得到带权评分。然 后，计算带权评分的总计TSo, 除以相似度总计Tsm, 以 Tsw/Tsm 作为排序标准，对这 些商品进行排序，然后推荐给用户 C 。比如，根据计算，商品103排名靠前，应该优先 推 荐 。

表5-4 **为用户C推荐商品**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| k-近邻用户 | 和C 相似度 | 商品101 评分 | 带权  评分 | 商品102 评分 | 带权  评分 | 商品103 评分 | 带权  评分 |
| 用户D | 0.98 | 3.4 | 3.332 | 4.4 | 4.312 | 5.8 | 5.684 |
| 用户E | 0.95 | 3.2 | 3.04 | / | 0 | 4.1 | 3.895 |
| 带权评分总计Tscor |  |  | 6.372 |  | 4.312 |  | 9.579 |
| 相似度总计Tsm | 1.93 |  |  |  |  |  |  |
| Tsom/Tsm  作为排序标准 |  |  | 3.30 |  | 2.23 |  | 4.96 |

资料来源：<http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudyl/index.html.>

用户C 获得的推荐，是从与他偏好相似的用户D,E 评价的商品里获得的。与用户C 相似度高的用户评分高的商品，将被优先推荐。

基于用户的协同过滤，需要找出用户的k 最近邻，可以根据用户评分矩阵计算和推 断。用户评分矩阵记录的是不同用户对不同的物品 (Item) 的评分。它的每一行对应一个 用户，它的每一列对应一个物品，<i,j> 单元记录的是用户i 对物品j 的评分R,。

用户评分矩阵的具体形式如表5-5所示。



①<http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy1/index.html.>

②<http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/index.html.>

**表5-5** **用户评分矩阵**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 物品1 | 物品2 | … | 物品n |
| 用户1 | Rn | R₁₂ | … | R₁n |
| 用户2 | R₂₁ | R₂2 | … | Rzn |
| : | … | : |  | … |
| 用户m | Rm: | Rm₂ | … | R… |

这里的评分表示用户对物品的偏好程度，评分越高表示用户越偏好该物品。以电商领 域为例，用户对商品的浏览、向朋友推荐、收藏、评论、购买等行为，都表现了用户对商 品的偏好程度。可以对这些行为进行量化，形成量化指标，然后对这些行为的量化指标进 行加权，计算最终评分。

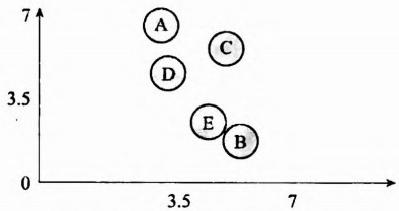
针对评分矩阵，基于用户的协同过滤推荐，是在每个用户对应的行向量上计算用户的 相关性。度量向量之间相似度的方法有很多，包括欧式距离、向量夹角、 Pearson 相关系 数等。Pearson 相关系数的计算公式比欧式距离的计算公式要复杂一些，但是在评分数据 不规范的情况下，Pearson 相关系数能够给出更好的结果。

我们通过如下的实例了解用户之间的相似度，这里用的是欧式距离。假设用户对各个 商品的评分如表5-6所示。

**表5-6** **用户对商品的评分**

|  |  |  |
| --- | --- | --- |
| 用户 | 商品1 | 商品2 |
| A | 3.32 | 6.51 |
| B | 5.78 | 2.62 |
| C | 3.59 | 6.29 |
| D | 3.42 | 5.78 |
| E | 5.19 | 3.11 |

我们绘制一个散点图，以商品1得分作为横坐标，以商品2得分作为纵坐标。用户 A,B,C,D,E 在坐标系上的分布如图5-10所示。由此可以看出，用户A,C,D 距离 较近，用户B,E 距离较近。



**图5-10** **用户对商品的评分的散点图**

从基本原理可以看到，该方法和基于人口统计学的推荐机制很类似。但是，基于人口 统计学的推荐机制只考虑用户的特征，基于用户的协同过滤机制，是在用户的历史偏好数 据上计算用户的相似度，它假设喜欢类似物品的用户可能具有相同或者相似的口味，这是 合理的。

(2)基于项目的协同过滤推荐。

基于项目的协同过滤推荐，是使用所有用户对物品或者内容的偏好信息，发现物品和

物品之间的相似度，然后根据用户的历史偏好信息，将类似的物品推荐给用户。以电商领 域为例，当需要对用户C 基于商品3进行商品推荐时，首先寻找商品3的k 近邻商品，也 就是和商品3相似的商品，比如商品3的k 近邻为商品4和商品5。然后计算商品4,5与 其他商品的相似度，并且进行排序，然后向用户C 推荐新的商品，也就是用户C 没有浏览 过或者购买过的商品。

比如，表5-7列出了用户C 购买过的商品4,5与其他商品(包括商品101,102, 103)的相似度。我们将用户C 对商品4,5的评分作为权重，乘以其他商品和商品4,5 的相似度，得到带权相似度。然后，计算相似度总计 Tsm, 除以评分总计 TSor, 以 Tsm/ Tsore作为排序标准，对商品101,102,103进行排序，然后推荐给用户C 。比如，根据计 算，商品103排名靠前，应该优先推荐。

**表5-7** **对用户C基于商品3进行推荐**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| k-近邻商品 | 评分 | 与商品10 相似度 | 带权 相似度 | 与商品102 相似度 | 带权 相似度 | 与商品103 相似度 | 带权 相似度 |
| 商品4 | 4.2 | 0.2 | 0.84 | 0.65 | 2.73 | 0.95 | 3.99 |
| 商品5 | 4.5 | 0.3 | 1.35 | 0.40 | 1.8 | 0.78 | 3.51 |
| 相似度总计Tsm |  |  | 2.19 |  | 4.53 |  | 7.5 |
| 评分总计Tsore | 8.7 |  |  |  |  |  |  |
| Tsm/Tsore  作为排序的标准 |  |  | 0.25 |  | 0.52 |  | 0.86 |

资料来源：<http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/index.html.>

用户C 获得的推荐是从与他购买过的商品相似度较高的商品中选出的。与用户C 评分 高的商品相似度高的商品被优先推荐。

针对评分矩阵，基于项目的协同过滤推荐，是在每个物品对应的列向量上，计算物品 的相关性。

从基本原理可以看到，该方法和基于内容的推荐，都是基于物品相似度进行推荐，但 是它们的相似度计算方法不一样。基于内容的推荐，仅仅基于物品本身的属性信息进行相 似度计算，基于项目的协同过滤推荐，则是从用户的历史偏好进行相似度计算。

如何在基于用户的协同过滤推荐与基于项目的协同过滤推荐之间做出选择，需要根据 应用场景的特点。比如，在电商领域， 一般来讲，物品的个数是远远小于用户的数量的， 而且物品的个数和相似度相对比较稳定，基于项目的机制比基于用户的机制更加适合，它 的实时性也更好。物品的相似度可以离线先算好，定期更新即可。在新闻推荐系统中，新 闻的个数(物品数)可能大于用户的数量，新闻更新也非常快，存在话题迁移，新闻的相 似度不稳定，这时候使用基于用户的协同过滤推荐算法，效果会更好。

**基于协同过滤的推荐机制是应用最为广泛的推荐机制。该方法的优势包括：(1)它无** **须对用户、物品进行严格的建模，它是领域无关的；(2)该方法支持用户发现潜在的兴趣** **偏好。**

同时我们也需要了解到，该方法存在若干问题：(1)该方法基于历史数据做出推荐， 对新用户和新物品存在“冷启动”问题；(2)推荐效果依赖于用户历史偏好数据的数据量 及其准确性；(3)少部分人的错误偏好，可能会对推荐的准确度产生很大的影响；(4)不 能照顾特殊偏好和品位的用户，不能给予精细的推荐。

5.2.7 kNN(k 近 邻 ) 算 法

kNN(k Near Neighbors) 算法是一种分类算法，它根据某个数据点周围的最近k 个 邻居的类别标签情况，赋予这个数据点一个类别。具体的过程是，给定一个测试数据点， 计算它与数据集中其他数据点的距离；找出距离最近的k 个数据点，作为该数据点的近邻 数据点集合；根据这k 个近邻所归属的类别，来确定当前数据点的类别。

比如，在图5-11中，采用欧式距离， k 的值确定为7,正方形表示类别一，三角形 表示类别二。现在要确定灰色方块的类别，图中的圆圈表示其k 最近邻所在的区域。在圆 圈里面，其他数据点的分类情况是，类别一有5个，类别二有2个。采用投票法分类，根 据多数原则，灰色数据点的分类确定为类别一。

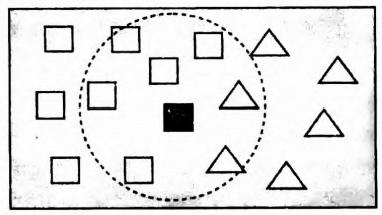
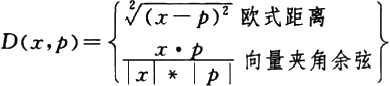
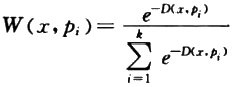


图5-11 kNN 算法实例

kNN 算法中，可用的距离包括欧式距离、夹角余弦等。 一般对于文本分类来说，用 夹角余弦计算距离(相似度),比欧式距离更为合适。距离越小(距离越近),表示两个数 据点属于同一类别的可能性越大。下面为距离公式 (x 为需要分类的数据点(向量),p 为 近邻数据点)。



当 k 个最近邻居确定之后，当前数据点的类别确定，可以采用投票法或者加权投票 法。投票法即根据少数服从多数的原则，近邻中哪个类别的数据点越多，当前数据点就属 于该类。加权投票法则根据距离的远近，对近邻的投票进行加权，距离越近权重越大，权 重为距离平方的倒数，最后确定当前数据点的类别。权重的计算公式为 (k 个近邻的权重 之和正好是1):



kNN 算法容易理解，也容易实现，无须进行参数估计，也无须训练过程，有了标注 数据之后，直接进行分类即可。kNN 算法可以对稀有的事件进行分类，比如客户流失预 测、欺诈侦测等。该算法也适用于多类别分类，也就是对象具有多个类别标签，比如某个 基因序列有多个功能， 一段文本有多个分类标签等。虽然具有这么多的优点， kNN 算 法 也有缺点，主要的缺点是该算法在进行数据点分类时计算量大，内存开销大，执行速度 慢。另外，该算法无法给出类似决策树的规则，结果的可解释性差。



kNN 算法中，k 值的选择非常重要。如果k 值太小，那么分类结果容易受到噪声数据 点影响；k 值太大，则近邻中可能包含太多其他类别的数据点。上述加权投票法可以降低 k 值设定不适当的一些影响。根据经验法则， 一般来讲k 值可以设定为训练样本数的平 方根。

kNN 分类算法的应用非常广泛，人们把它应用到协同过滤推荐 (Collaborative Filte- ring) 、手写体识别 (Digit Recognition) 等领域。

5.2.8 朴素贝叶斯 (Naive Bayes) 算 法

贝叶斯分类是一类分类算法的总称，它们都以贝叶斯定理为基础。下面我们首先介绍 贝叶斯定理，然后结合实例讨论朴素贝叶斯分类，它是贝叶斯分类中最简单的一种方法。

1. 贝叶斯定理

P(B|A) 表示在事件A 已经发生的前提下，事件B 发生的概率，称为事件A 发生情 况下，事件 B 发生的条件概率。

在实际应用中经常遇到这样的情况，我们可以很容易计算 P(A|B), 但 是P(B|A) 则 很难直接得出。我们要计算的目标是P(B|A), 贝叶斯定理为我们从P(A|B) 计算P(B|

A) 提供了一种途径。贝叶斯定理的具体形式为： 。该公式的 证明过程，可以参考概率论的相关教材。

2. 朴素贝叶斯分类

朴素贝叶斯分类是运用上述贝叶斯定理，并且假设特征属性 (Feature Attribute) 是 条件独立的一种分类方法，即朴素贝叶斯分类器假设样本的每个特征与其他特征都不 相关。

假设我们有如下的分类问题(在下面的描述中涉及N 个样本，下标用s,K 个类，下 标用i,M 个特征属性，下标用j, 某个属性a;有L;个划分，下标为l):

(1)假设x={a₁,a₂,…,am} 为一个待分类项(即一个向量),a₁,a₂,…,am 为 x 的特征属性。

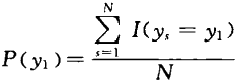
(2)类别集合C={y₁,y₂,…,ye}, 总共有K 个分类。

(3)我们要判断x 属于哪个分类，于是计算P(y₁ |x),P(y₂ |x),…,P(yk |x)。

(4)如果P(y; |x)=max{P(y₁ Ix),P(y₂ |x),…,P(yk |x)},那么x∈y;。也就是 P(y; |x),i=1,…,K, 哪个最大，x 就最可能属于哪个类别。

问题就转换成计算P(y₁lx),P(y₂|x),…,P(y₆|x), 也就是x 属于各个类别的概率。

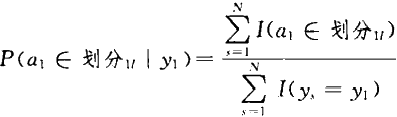
其具体计算过程如下：

(1)创建一个已经知道其分类的待分类项集合，这个集合称为训练样本集，样本集包 含N 个样本。,式中，s 为样本下标，I为一个指示函数，若括号内 成立，则计1,否则为0。

(2)根据训练样本集，统计得到各个类别下各个特征属性的条件概率估计。也就是计 算 P(a₁I y₁),P(a₂I y₁),…,P(amI y₁);P(aiI y₂),P(a₂I y₂),…,P(am I y₂);…; P(a₁ Iyk),P(a₂ I yk),…,P(amI yk)等 ，a₁,a₂,…,am 是M 维特征向量的各个维度。

当计算 P(a₁I y₁)元素时，就是计算在分类为y₁ 的样本上 .a₁ 的值域的各个划分的概 率，比如y₁ 类样本有100个， a1 的值域划分成3个子划分，各个子划分的样本分别包含 10,70 . 20个样本，那么P(a₁∈ 划分I y₁)=0.1.P(a₁ ∈ 划分12Iy₁)=0.7.P(a₁∈

划分13I y₁)=0.2。总结起来，



式 中 .l∈1.….L(L₁=3) 为 a!的值域的划分的个数。其他P(a,I y;)的根据同样道理.进 行计算。

(3)假设各个特征属性是条件独立的.根据贝叶斯定理，有如下的推导：



式中，分母对于所有类别为常数，我们只需将分子最大化即可。由于各个特征属性是条件 独立的，所以



整个朴素贝叶斯分类分为三个阶段，分别是准备阶段、训练阶段和应用阶段。

(1)准备阶段的任务是为朴素贝叶斯分类做必要的准备。主要工作是根据具体情况， 确定特征属性，并对每个特征属性进行适当划分、然后对一部分待分类项进行人工分类， 形成训练样本集合。第一个阶段的数据质量对整个过程将有重要影响，分类器的质量很大 程度上由特征属性、特征属性划分以及训练样本质量决定。

(2)训练阶段的主要任务是生成分类器。主要工作是计算每个类别在训练样本中的出 现频率(上式中的P(y)). 以及每个特征属性划分对每个类别的条件概率估计(上式中的 P(a;I y;)),并且记录结果。

(3)应用阶段的主要任务是使用分类器.对待分类项进行分类.也就是对新数据进行 分类。

3. 朴素贝叶斯分类实例

文献①给出了一个实例。通过朴素贝叶斯分类来检测 SNS(Social Networking Serv- ice) 社区中的不真实账号。在SNS 社区中.不真实账号是一个普遍存在的问题。 SNS 社 区的运营商希望检测出这些不真实账号，从而加强对 SNS 社区的监管和治理。使用人工 检测方法，需要耗费大量的人力.效率低下。如果能够设计和使用某种自动检测方法。将 大大提高工作效率。

这个工作的目的是根据账号的一些属性.把它们划分成真实账号和不真实账号两类。 那么目标分类集合C={0.1}.0 表示真实账号. 1表示不真实账号。使用朴素贝叶斯分 类方法，对账号进行分类的具体过程如下：

(1)确定特征属性以及属性值域的划分。在实际应用中.特征属性的数量是很多的， 划分也会比较细致。在这里主要目的是对朴素贝叶斯分类方法进行说明.仅仅使用少量的

①<http://www.cnblogs.com/leoo2sk/archive/2010/09/17/naive-bayesian-classifier.html.>

特征属性以及较粗的划分。

在本实例中使用三个属性， a₁ 是日志数量/注册天数， a₂ 是好友数量/注册天数， a₃ 表 示是否使用真实头像。每个账号的这三个属性，都可以从系统中查询出来，或者计算出 来。这三个属性的值域的划分如下，a:{ [一∞,0.05]. (0.05.0.2),[0.2,+∞)},a₂ :{ ( 一 ∞, 0.1],(0.1.0.8),[0.8,+∞]},a₃ :{0.1},0 表示未使用真实头像，1表示使用真实头像。

(2)获取训练样本，可以使用运维人员曾经检测的1万个账号作为训练样本。

(3)利用训练样本中真实账号和不真实账号，计算各个类别的概率。比如，真实账号 P(c=0)=8900/10000=0.89, 不真实账号P(c=1)=1100/10000=0.11。

(4)计算每个类别下，各个特征属性划分的概率。比如：

P(a₁ ∈ (一∞,0.05) |c=0)=0.3,P(a₁ ∈(0.05,0.2) |C=0)=0.5,P(a₁ ∈[0.2, 十∞) |c=0)=0.2

P(a₁ ∈ (一∞,0.05) |c=1)=0.8.P(a₁ ∈(0.05,0.2) |C=1)=0.1,P(a₁ ∈[0.2, 十∞) |c=1)=0.1

P(a₂ ∈ (一 ∞,0.1) |c=0)=0.1,P(a₂ ∈(0.1,0.8) |C=0)=0.7,P(a₂ ∈[0.8,

+∞)|c=0)=0.2

P(a₂ ∈ [一 ∞,0.1] |c=1)=0.7.P(a₂ ∈(0.1,0.8) |C=1)=0.2,P(a₂ ∈[0.8, 十∞) |c=1)=0.1

P(a₃=0|C=0)=0.2,P(a₃=1|C=0)=0.8

P(a₃=0 |C=1)=0.9,P(a₃=1|C=1)=0.1

(5)使用分类器进行分类。假设现在有一个账号 x, 该账号的属性 a₁=0.11,a₂=

0.22,a₃=0, 那 么

P(C=0)P(x|C=0)=P(C=0)P(a₁∈(0.05,0.2)|C=0)P(a₂∈(0.1,0.8)|

C=0)P(a₃=0 |C=0)=0.89×0.5×0.7×0.2=0.0623

P(C=1)P(x|C=1)=P(C=1)P(a₁∈(<0.05.0.2>)|C=1)P(a₂∈(0.1,0.8)|

C=1)P(a₃=0 |C=1)=0.11×0.1×0.2×0.9=0.00198

由于前者大于后者 .所以该账号属于C=0 类的概率更大，即该账号为真实账号 类别。

4. 属性值为连续值的处理方法

当某个特征属性为连续的值时，通常假设其服从正态分布，即



即 *P(a;I y;)=g(a;,no)*

只要计算出训练样本中各个类别中这个特征属性的值域划分的均值和标准差，代入上 述公式.就可以得到需要的估计值。

这里举另外一个实例.该实例来自维基百科① .是一个处理连续变量的例子。

下面是一组人类身体特征的统计资料(见表5-8)。



① <http://en.wikipedia.org/wiki/Naive_Bayes#Sex_classification.>

**表5-8** **身体特征统计资料**

|  |  |  |  |
| --- | --- | --- | --- |
| x | | | y |
| 身高(英尺)x ·ai | 体重(磅)x ·a2 | 脚掌(英寸)x ·as | 性别 |
| 6 | 180 | 12 | 男 |
| 5.92 | 190 | 11 | 男 |
| 5.58 | 170 | 12 | 男 |
| 5.92 | 165 | 10 | 男 |
| 5 | 100 | 6 | 女 |
| 5.5 | 150 | 8 | 女 |
| 5.42 | 130 | 7 | 女 |
| 5.75 | 150 | 9 | 女 |

现在知道某人身高6英尺、体重130磅、脚掌8英寸，请问该人是男是女?

根据朴素贝叶斯分类器的原理，我们需要计算， P( 身 高 | 性 别 ) ×P( 体 重 | 性 别 ) × P(脚掌 | 性别)×P (性别)。现在的困难在于由于身高、体重、脚掌都是连续变量，不能 采用离散变量的方法计算概率，而且由于样本太少，也无法分成区间计算。这时可以假设 男性和女性的身高、体重、脚掌都是正态分布，通过样本计算出均值和方差，也就是得到 正态分布的密度函数，具体如表5-9所示。此外，我们认为 P ( 男 ) =P(女)=0 . 5。

**表5-9** **正态分布的密度函数**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gender | mean  (height) | variance  (height) | mean  (weight) | variance  (weight) | mean  (foot size) | variance  (foot size) |
| male | 5.855 | 3.5033e-02 | 176.25 | 1.2292e+02 | 11.25 | 9.1667e-01 |
| female | 5.4175 | 9.7225e-02 | 132.5 | 5.5833e+02 | 7.5 | 1.6667e+00 |

有了密度函数，就可以把值代入，算出某一点的密度函数的值。比如，男性的身高是 均值5.855、方差0.035的正态分布。男性的身高为6英尺的概率的相对值等于1.5789。



最后，我们计算得到：

P (身 高 = 6 | 男 ) ×P ( 体 重 = 1 3 0 | 男 ) ×P ( 脚 掌 = 8 | 男 )

×P (男)=6 . 1984×e⁻⁹

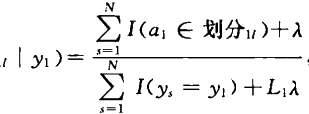
P(身高 = 6 |女) × P (体 重= 1 3 0 |女) × P (脚 掌= 8 |女)

×P (女)=5 . 3778×e⁴

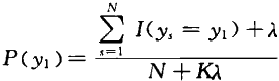
可以看到，女性的概率比男性要高出将近10000倍，因此判断该人为女性。

5.Laplace 校准

当出现P(a;I y;)=0 等情况，即某个类别下某个特征属性没有出现时，分类器的质 量将大大降低。为了解决这个问题， 一般引入Laplace 校准。

比如，P(a₁ ∈ 划分 ,L 为 a₁ 的值域的划分数量，如

果a₁ 的值域划分成3个子域，那么L₁ 就是3。其他P(a;I y)同理计算。当训练样本集合 足够大时，不会对结果产生影响，同时解决了P(a,I y;)=0 的问题。此外，各个类别的



概率的计算也需要调整，比如，P(y₁) 的计算公式调整为 为类别数量，其他各类别概率同理计算。

,K

**5.2.9 AdaBoost 算** **法**

Boosting算法系列的思想来自 PAC 可学习性 (Probably Approximately Correct Learnability) 理论。该理论研究什么时候一个问题是可被学习的，以及可学习问题的具体 学习方法。Valiant 和Kearns 首次提出了PAC 学习模型中弱学习算法和强学习算法的等 价性问题，即任意给定仅比随机猜测稍微好一点的弱学习算法，是否可以将其提升为强学 习算法?如果二者等价，那么只需找到一个比随机猜测略好的弱学习算法，就可以将其提 升为强学习算法而不必寻找很难获得的强学习算法。

AdaBoost 是英文 “Adaptive Boosting” (适应性提升)的缩写，由 Yoav Freund 和 Robert E.Schapire 在1995年提出，回答了上述问题。他们利用 AdaBoost 算法，把多个 不同的决策树用一种非随机的方式组合起来，表现出惊人的性能。首先，决策树的准确率 大大地提高了，可以与SVM 媲美。其次，运行速度快且基本不用调参数。最后，该组合 分类器几乎不产生过拟合 (Over Fitting) 现象。

Boosting算法是一种把多个分类器整合为一个分类器的方法。在 Boosting算法产生之 前还出现过两种比较重要的类似方法，Bootstrapping 方法和 Bagging 方法，读者可以参考 相关资料。

**1.** **算法思想**

AdaBoost 是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器，即 弱分类器，然后把这些弱分类器集合起来，构造一个更强的最终分类器。算法本身是通过 改变数据分布实现的，它根据每次训练集之中的每个样本的分类是否正确，以及上次的总 体分类的准确率，来确定每个样本的权值。将修改了权值的新数据送给下层分类器进行训 练，然后将每次训练得到的分类器融合起来，作为最后的决策分类器。

AdaBoost 的适应性在于前一个基本分类器分错的样本会得到加强，加权后的全体样 本再次用来训练下一个基本分类器。同时，在每一轮中加入一个新的弱分类器，直至达到 某个预定的足够小的错误率或达到预先指定的最大迭代次数。

具体来讲，整个AdaBoost 迭代算法包含3个主要步骤：

(1)初始化训练数据的权值分布。如果有M 个样本，则每一个训练样本最开始时都 被赋予相同的权值：1/M。

(2)训练弱分类器。在训练过程中，如果某个样本点已经被准确地分类，那么在构造 下一个训练集中它的权值就被降低；相反，如果某个样本点没有被准确地分类，那么它的 权值就得到提高。在第t 轮训练结束后，根据得到的弱分类器h, 的性能，计算该分类器对 应的权值α,并由h, 在训练集上的分类结果对权重向量W,→W+1 进行更新。接着，权值 更新过的样本集用于训练下一个分类器，整个训练过程如此迭代地进行下去。

(3)将各个训练得到的弱分类器组合成强分类器。各个弱分类器的训练过程结束后，



加大分类误差率小的弱分类器的权重，使其在最终的分类函数中起较大的决定作用，降低 分类误差率大的弱分类器的权重，使其在最终的分类函数中起较小的决定作用。换言之， 误差率低的弱分类器在最终分类器中占的权重较大，否则较小。

具体的算法流程如下：

|  |
| --- |
| 给定：(x₁,y₁),…,(xm,ym),x;∈X,y;∈Y={-1,+1}  1. 为训练集中每个样本，初始化权重W₁ (i)=1/m。  2. 从 t=1,…,T 迭代  2.1 使用样本分布W, 对弱分类器h, 进行训练  2.2 计算弱分类器h:X→ {-1,+1} 的带权分类误差 E,=∑W,(i)I[h,(x;)≠y;]  2.3 计算弱分类器对应的权重  2.4 更新样本权重      其中，SUM(Wt)是一个规范化因子(从而使得 De+1是一个分布)  3. 得到 T 个不同的弱分类器及其对应的权重，输出最终的分类器    利用这个分类器，对样本数据进行分类。 |

2.2步，分类器h, 的性能度量，即该分类器在训练集上的结果，通过计算该分类器在 训练集上的带权分类误差来计算。带权分类误差是指将待分类的样本包含的权重(初始化 的样本权重或者前一次迭代以后调整的权重),结合该数据集上的分类误差，得到分类器 在该数据集上的一个考虑样本权重的分类误差。计算公式为：

E,=∑W,(i)I[h,(x;)≠y.]

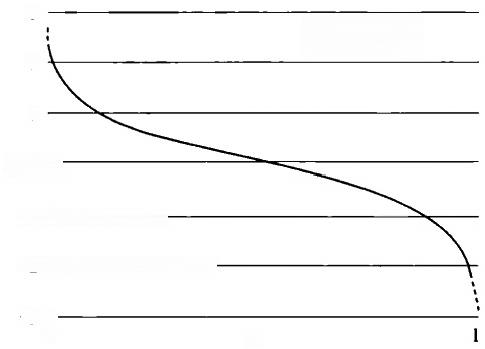
式中，ε,为第t 个弱分类器的带权分类误差值； w₂ (i) 为 第t 次更新后样本i 的权重； h,(xi)为使用第1个弱分类器对样本i的分类结果；y; 为样本i的真实标签；I[h,(x;)≠y;]

是一个指示函数，它的值是

2.3步，弱分类器 h, 对应的权重α,与其带权分类误差有关，计算公式为：α,=

 (见图5-12)。通过绘制分类器的权重函数的图像，我们可以看到，带权分类 误差的范围是[0,1]。弱分类器的权重与其对应的带权分类误差呈反比关系，也就是带 权分类误差越小，该分类器对应的权值越大，或相反。





3

2 a-₂n ( 二 )

1

a,0—

-1 ---

—2 —

-3 一

0 E,

图5- 12  函数图像

2.4步，更新样本权值 。该式定义了计算 弱分类器h, 对应的权值α后，对样本i 的权重更新过程。如果该分类器在该样本上分类正 确，则降低该样本的权值；如果分类错误，则提高该样本的权值。公式前半部分主要用于 对整个权重向量进行规范化处理，以使其和为1。

从上述过程可以看出，每轮训练结束后， AdaBoost 算法对样本的权重进行调整，该 调整的结果是越到后面被错误分类的样本权重会越高。于是，到后面单个弱分类器为了达 到较低的带权分类误差，都会把样本权重高的样本分类正确。最终的结果是，虽然每个弱 分类器可能都有分错的样本，然而整个AdaBoost 框架却能保证对每个样本进行正确分类。

2.AdaBoost 算法的特点

AdaBoost 是一种具有很高精度的分类器，其算法具有如下特点：

(1)可以使用各种方法构建子分类器. AdaBoost 算法提供对其进行组合以及提升的 框架。

(2)当使用简单分类器时，计算出的结果是可以理解的。

(3)弱分类器构造极其简单，无须做特征筛选。

(4)AdaBoost 算法简单，不用调整分类器，不会导致过拟合。

3.AdaBoost 算法的应用

AdaBoost 算法的应用场景包括：

(1)用于二值分类或多分类的应用场景。

(2)用于特征选择 (Feature Selection)。

(3)无须变动原有分类器，而是通过组合出新的分类器，提升分类器的性能。

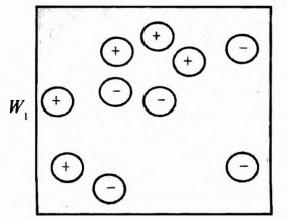
4.AdaBoost 实例

我们通过如下的实例来深人把握 AdaBoost 算法的思想及其产生的效果。

(1)如图5 - 13所示， W₁ 表示样本的初始权重分配，数据点包含两类数据，分别用 “+”号和“一”号表示。在AdaBoost 算法运行过程中，我们使用水平或者垂直的直线作为 分类器来进行分类。算法最开始给了一个均匀分布D 。因此，h 里的每个点的权重是0.1。

(2)利用第一个分类器进行划分，有三个数据点划分错了，根据误差公式，计算得到 带权的误差为： E=(0.1+0.1+0.1)=0.3 。 第一个分类器的权重a₁ 为0.42。根据算法

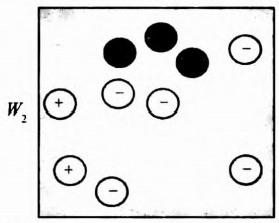
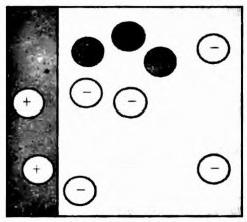




**图5** **-** **13** **AdaBoost实例**

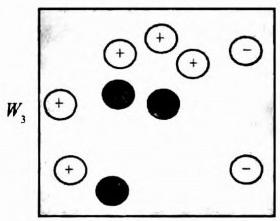
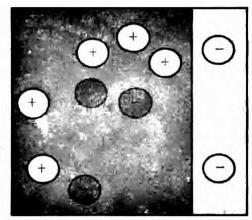
要求，把分错的数据点的权值变大(图5-14中，圆圈的灰度变深表示权值变大，因为分 类器对其分类错误),得到新的权重分布。

至此，根据分类的正确率得到 一 个新的样本权重分布W₂, 一 个 子 分 类 器h。



*h.* ε,=0.3 a₁=0.42

**图5** **-** **14** **AdaBoost第1次迭代**

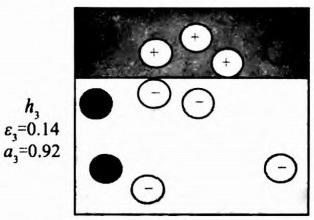
(3)进行第2次迭代，根据分类的正确率，得到 一 个新的样本权重分布 W₃ 以 及 一 个 子分类器h₂ ( 见 图 5 - 1 5 ) 。

h₂

e₂=0.21 a₂=0.65

**图5** **-** **15** **AdaBoost第2次迭代**

(4)计算最后一个分类器的错误率和权重，得到最后一个分类器的权重为h₃ (见图5 - 16)。



**图** **5** **-** **1** **6** **AdaBoost** **第3次迭代**



(5)整合所有子分类器，即对其进行加权求和。从结果中看，即使简单的分类器，组 合起来也能获得很好的分类效果(见图5-17)。

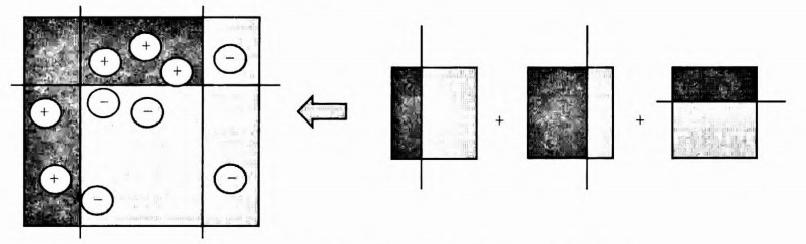


图5-17 hral=sign(0.42×h₁+0.65×h₂+0.92×h₃)

资料来源：<http://erinshellmangithub.io/data-mining-starter-kit/#/108.>

5.2.10 线性回归、Logistic 回 归

1. 线性回归与多元线性回归

回归分析是应用广泛的统计分析方法，用于分析事物之间的相关关系。其中一元线性 回归模型指的是只有一个解释变量的线性回归模型，多元线性回归模型则是包含多个解释 变量的线性回归模型。解释变量就是自变量，被解释变量则是因变量。回归模型就是描述 因变量和自变量之间依存的数量关系的模型。

一元线性回归模型具有y=ax+b 的简单形式，a 称为自变量x 的系数，b 称为截距， y=ax+b 对应到图形则是二维平面上的一条直线。扩展到多元线性回归模型，其形式为

,方程中包含n 个自变量x₁,x₂, … ,xn, 其系数分别是a₁,a₂,…,an。

文献①给出了一个多元线性回归实例，我们通过该实例，具体了解多元线性回归模型 的建立和应用。现有12名大学一年级女生的身高、体重和肺活量数据，如表5-10所示。 现在我们希望在这些数据上建立一个回归模型。这个模型有两个目的：一个是解释这些数 据，也就是肺活量和身高、体重有什么关系；另一个是进行预测，假设我们获得另外一个 女生的身高、体重数据，我们希望用这个模型预测出她的肺活量应该是多少。

**表5-10** **12名大学一年级女生的身高、体重以及肺活量**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 身高(cm) | 体重(kg) | 肺活量(L) |
| 1 | 161 | 42 | 2.55 |
| 2 | 162 | 42 | 2.2 |
| 3 | 165 | 46 | 2.75 |
| 4 | 162 | 46 | 2.4 |
| 5 | 166 | 46 | 2.8 |
| 6 | 167 | 50 | 2.81 |
| 7 | 165 | 50 | 3.41 |
| 8 | 166 | 50 | 3.1 |



①<http://www.sohu.com/a/30186905_216927.>



续前表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 身高(cm) | 体重(kg) | 肺活量(L) |
| 9 | 168 | 52 | 3.46 |
| 10 | 165 | 52 | 2.85 |
| 11 | 170 | 58 | 3.5 |
| 12 | 168 | 58 | 3 |

我们把身高作为第一个自变量x, 体重作为第二个自变量 x₂, 肺活量则作为因变量 y 。我们要建立的方程为y=a₁x₁+a₂x₂+b 。 经过计算(使用最小二乘法估计),得出

b=—0.5657,a₁=0.005017,a₂=0.05406 于是，多元线性回归方程为：

y=0.005017x₁+0.05406x₂—0.5657

a₁=0.005017 表示在 x₂ 即体重不变的情况下，身高每增加1cm, 肺活量增加

0.005017L。

建立多元线性回归模型的目的是解释数据以及进行预测。比如，现在遇到一个新的大 学一年级女生.身高166cm, 体重46kg, 把这两个数据代入上述方程，得到y=2.75, 表 示对于这样身高和体重的女生，估计的肺活量为2.75L。

2. 多元线性回归模型的检验

多元线性回归模型建立以后需要从几个角度进行检验，以了解模型的解释能力和预测 能力。这些检验包括拟合优度检验、回归方程显著性检验、回归系数的显著性检验。

我们把因变量的总变差 (SST) 分解成自变量变动引起的变差 (SSR) 和其他因素造成 的变差 (SSE)。 用数学语言表达为：

SST=∑(y—y)²=∑(-y²+∑(y-9)²=SSR+SSE 式中，y 为样本均值； 为模型预测值；y 为因变量的实际值。

(1)拟合优度检验。回归方程的拟合优度指的是回归方程对样本的各个数据点的拟合 程度。拟合优度的度量一般使用判定系数R², 它是在因变量的总变差中由回归方程解释的 变动(回归平方和)所占的比重。R² 越大，方程的拟合程度越高。R² 的计算公式为：



当一个多元线性回归模型的判定系数接近1.0,说明其拟合优度较高。

(2)回归方程显著性检验。回归方程的显著性检验. 目的是评价所有自变量和因变量 的线性关系是否密切。常用F 检验统计量进行检验. F 检验是对模型整体回归显著性的检 验。F 统计量的计算公式为：



式中，n 为样本容量；k 为自变量个数。

F 检验的原假设 (H₀) 为，自变量和因变量的线性关系不显著;备择假设 (H₁) 为， 自变量和因变量的线性关系显著。在给定的显著性水平(一般选0.05)下，查找自由度为 (k,n—k—1) 的 F 分布表，得到相应的临界值Fa。 如果上述计算公式算得的F>Fa. 那 么拒绝原假设，回归方程具有显著意义，回归效果显著,否则，F<F, 那么接受原假设， 回归方程不具有统计上的显著意义，回归效果不显著。我们也可以通过概率P 值来进行判 断，概率P 值表示H。成立的可能性有多大。当P<0.05, 表示概率极低，那么拒绝原假



设，回归方程具有显著意义，回归效果显著,否则， P>0.05, 在给定显著性水平下，我 们不能轻率地拒绝原假设，而是接受它，即自变量和因变量的线性关系不显著。

(3)回归系数的显著性检验。使用t 检验，分别检验回归模型中的各个回归系数是否 具有显著性，以便使模型中只保留那些对因变量有显著影响的因素。 t 检验是对单个解释

变量回归系数的显著性检验。回归系数i的t 检验统计量为,其中SB 表示回归系数 β的标准误差，具体计算方式可以参考相关资料。

t 检验的原假设(H₀) 为 ，a, 的值为0,即对应变量x;的系数为0,该变量无须进人方 程；备择假设(H₁) 为 ，a; 的值不为0,即对应变量x; 的系数不为0,该变量需要进入方程。 给定显著性水平a (一般选0.05),查找自由度为n—k—1 的 t 分布表，得到临界值t, 如 果t₁>ta, 拒绝原假设，回归系数ai 与0有显著差异，对应的自变量x; 对因变量y 有解释 作用；否则 t;<ta, 接受原假设，回归系数a;与0没有显著差异，对应的自变量xi对因变 量y 没有解释作用。我们也可以通过概率P 值来进行判断，概率 P 值表示H。成立的可能 性有多大。当P<0.05, 表示概率极低，那么拒绝原假设，a; 对应的变量x; 系数不为0, 该变量需要进入方程；否则，P>0.05, 在给定显著性水平下，我们不能轻率地拒绝原假 设而是接受它，即a;对应的变量xi 系数为0,该变量无须进入方程。

如果某个回归系数的t检验显示该自变量对因变量没有显著解释作用，这时应该从回归 模型中剔除这个变量，重新建立更为简单的回归模型，或者增加其他更加相关的自变量。

3. 共线性的检测

共线性指的是自变量之间存在较强的线性相关关系。这种关系如果超越了因变量与自 变量的线性关系，那么回归模型就不准确了。在多元线性回归模型中，共线性现象无法避 免，只要不太严重就可以了。

检测回归方程是否存在严重的共线性，可以分别计算每两个自变量之间的判定系数 r², 然后和回归方程的拟合优度检验的判定系数做比较，如果r²>R² 或者非常接近R², 那 么存在严重的共线性，必须想办法降低共线性的影响。

另外一种检测方法是计算自变量之间的相关系数矩阵的特征值的条件数k=λ₁/λp(λ1 为最大的特征值， λp 为最小的特征值)。如果k<100, 表示自变量之间不存在严重的共线 性；如果100≤k≤1000, 那么自变量之间存在较强的共线性；如果k>1000, 则自变量 之间存在严重的共线性。降低共线性的办法主要是转换自变量的取值，比如变绝对数为相 对数或者平均数，或者更换成其他的自变量。

4.自变量筛选法

在多元线性回归中，存在一个自变量选择的问题，因为并不是所有的自变量都对因变 量有解释作用。比如，在上述实例中，通过身高、体重(自变量)和肺活量(因变量)建 立的回归模型中，我们引入一个血压数据，有可能和肺活量没有什么关系。自变量间可能 存在较强的线性关系，即共线性。因此，不能把所有的变量全部引入方程。变量选择的方 法包括前向筛选法、后向筛选法和逐步筛选法三种。

(1)前向筛选法 (Forward) 。 即自变量不断进入回归方程的过程。首先，选择与因变 量具有最高相关系数的自变量进入方程并进行各种检验。其次，在剩余的自变量中寻找偏 相关系数最高的变量，进入回归方程，并进行检验。反复上述步骤，直到没有可进入方程 的自变量为止。回归系数检验的概率P 值小于Pn(0.05), 才可以进入方程。

(2)后向筛选法 (Forward) 。 即自变量不断剔除出回归方程的过程。首先，将所有自 变量全部引入回归方程。其次，在一个或多个t 值不显著的自变量中，将t值最小的那个

变量剔除出去，并重新建立方程和进行检验。回归系数检验 P 值大于P(0.10), 则剔除 出方程。如果新方程中所有变量的回归系数t值都是显著的，则变量筛选过程结束，否 则，重复上述过程，直到没有变量可剔除为止。

(3)逐步筛选法 (Stepwise)。是前向筛选法和后向筛选法的结合。前向筛选法只对 进入方程的变量的回归系数进行显著性检验，对已经进入方程的其他变量的回归系数不再 进行显著性检验，也就是说，变量一旦进入方程就不会被剔除。随着变量的逐个引进，由 于变量之间存在着一定程度的相关性，已经进入方程的变量，其回归系数不再显著,因此 会造成最后的回归方程可能包含不显著的变量。逐步筛选法则在变量选择的每一个阶段， 都考虑剔除一个变量的可能性。

我们使用另外一个多元线性回归实例①,来说明如何进行变量选择和模型检验。有研 究人员收集了1998—2008年上海市城市人口密度(人/平方公里)、城市居民人均可支配 收入(元)、五年以上平均年贷款利率(%)和房屋空置率(%)等作为变量，在此基础 上通过多元线性回归，预测商品房平均售价(元/平方米)的变化。他们通过 SPSS 建立 回归模型，采用逐步变量筛选法对自变量进行筛选。

表5-11为引入/剔除变量表，从表中可以看出，模型首先引入城市人口密度，接着 引入城市居民人均可支配收入，没有变量被剔除。 SPSS 建立了两个模型，分别是模型 a 和模型 b。模型a 使用城市人口密度作为自变量，商品房平均售价作为因变量。模型b 使 用城市人口密度和城市居民人均可支配收入作为自变量，商品房平均售价作为因变量(比 模型 a 多了一个自变量)。

**表5-11** **引入/剔除变量表(Variables** **Entered/Removed)**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Variables Entered | Variables Removed | Methods |
| 1 | 城 市 人 口 密 度  (人/平方公里) |  | Stepwise(Criteria:Probability of F to enter<= 0.050.Probability of F to Remove>=0.100) |
| 2 | 城市居民人均可 支配收入(元) |  | Stepwise(Criteria:Probability of F to enter <=0.050,Probability of F to Remove>=0.100) |

注 ：Dependent Variable 为商品房平均售价(元/平方米)。

表5-12为模型汇总表，展示了模型的判定系数等参数。从判定系数可以看出，模型 a 和模型b 的 R² 系数都是1.0,回归方程的拟合优度很好。

**表5-12** **模型汇总表(Model** **Summary)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | R | R² | Adjusted R² | Std.Error of the Estimate | **Durbin-Watson** |
| 1 | 1.000° | 1.000 | 1.000 | 35.187 |  |
| 2 | 1.000 | 1.000 | 1.000 | 28.351 | 2.845 |

注：模型 a,Predictors 包 括Constant. 城市人口密度(人/平方公里)。

模型b,Predictors 包括Constant. 城市人口密度(人/平方公里),城市居民人均可支配收入(元)。

Dependent Variable为商品房平均售价(元/平方米)。

表5-13为方差分析表，从中可以看出整个回归方程是否显著。可以看到，模型 a 的 F 统计量为30938.620,概率P 值(第五列)为0 .00,模型b 的 F 统计量为23832.156. 概率 P 值为0.00。在显著性水平0.05下，两个模型的回归方程都显著,即城市人口密 度、城市居民人均可支配收入，都和商品房平均售价有线性关系。



①<http://www.doc88.com/p-7714452513095.html.>

**表5-13** **方差分析表(ANOVA:Analysis** **of** **Variance)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | | Sum of Squares | df | Mean Square | F | Sig. |
| 1 | Regression | 38305583.51 | 1 | 38305583.51 | 30938.62 | 0.000# |
| Residual | 11143.039 | 9 | 1238.115 |
| Total | 38316726.55 | 10 |  |
| 2 | Regression | 38310296.53 | 2 | 19155148.26 | 23832.156 | 0.000 |
| Residual | 6430.018 | 8 | 803.752 |
| Total | 38316726.55 | 10 |  |

注：模型a.Predictors 包括Constant, 城市人口密度(人/平方公里)。

模型 b,Predictors 包括Constant, 城市人口密度(人/平方公里),城市居民人均可支配收入(元)。 Dependent Variable 为商品房平均售价(元/平方米)。

表5-14为回归系数分析表.该表显示了各个模型针对每个系数的t 统计量及其概率 P 值。从表中可以看出，模型b 的城市人均可支配收入的概率 P 值较大，为0.042,但是 仍然小于0.05。两个模型的其他系数的概率 P 值都接近0.00,说明这两个模型的所有自 变量都对因变量有解释作用，必须保留在回归方程里。

**表5-14** **回归系数分析表(Coefficients)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | | Unstandardized  Coeficients | | Standardized  Coeficients | t | Sig. | Collinearity  Statistics | |
| B | Std.Error | Beta | Tolerance | VIF |
| 1 | Constant | 1652.246 | 24.137 |  | 68.454 | 0.000 |  |  |
| 城市人口密度  (人/平方公里) | 1.072 | 0.006 | 1.000 | 175.894 | 0.000 | 1.000 | 1.000 |
| 2 | Constant | 1555.506 | 44.432 |  | 35.009 | 0.000 |  |  |
| 城市人口密度  (人/平方公里) | 1.02 | 0.022 | 0.951 | 46.302 | 0.000 | 0.050 | 20.126 |
| 城市居民人均可 支配收入(元) | 0.017 | 0.007 | 0.050 | 2.422 | 0.042 | 0.050 | 20.126 |

注 ：Dependent Variablc 为商品房平均售价(元/平方米)。

表5-15为共线性分析表，显示了共线性检验的特征值以及条件指数。 Eigenvalue 为 各个模型的特征值，Condition Index为条件指数。如果条件指数小于30.表明不存在共 线性；在30～100之间.表明存在一定程度的共线性，但不会对模型的回归与解释产生影 响；如果高于100,则表明存在严重的共线性。在这个实例中，最大的条件指数为 30.736,表明虽然存在共线性，但是不会影响模型的解释能力。

**表5-15** **共线性分析表(Collinearity** **Diagnostics)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Dimension | Eigenvalue | Condition Index | Variance Proportions | | |
| constant | 城市人口密度  (人/平方公里) | 城市居民人均 可支配收入(元) |
| 1 | 1 | 1.898 | 1.000 | 0.05 | 0.05 |  |
| 2 | 0.102 | 4.319 | 0.95 | 0.95 |  |
| 2 | 1 | 2.891 | 1.000 | 0.00 | 0.00 | 0.00 |
| 2 | 0.106 | 5.213 | 0.21 | 0.03 | 0.00 |
| 3 | 0.003 | 30.736 | 0.78 | 0.97 | 1.00 |

注 ：Dependent Variable为商品房平均售价(元/平方米)。

5. 类别变量 (Category Variable) 处理办法

回归方程是一个描述数量关系的方程，但是在回归方程建立的过程中，某些变量并不

具有一个数值而是具有某种类别，也就是这些变量不是数值型变量而是类别型变量。比如 性别，不是取一个数值而是有“男”/“女”两个类别。

对于这样的类别变量， 一般采用给各个类别一个数值，把它转换成一个数值型变量。 比如，如果该变量的取值为“男”,转换成数值1,如果该变量的取值为“女”,转换成数 值0。通过把类别变量(定性变量)转换成数值变量，就可以在这些变量上按照常规方法 建立回归模型。

6. 使用回归模型进行预测

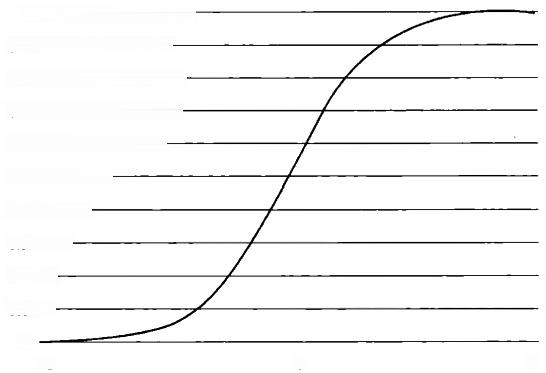
利用回归模型进行预测时，解释变量的值最好不要离样本数据的值域范围太远。主要 原因有两个：首先，预测点离样本平均值x 越远，则被解释变量(因变量)的预测误差越 大；其次，在样本所在的值域之外，变量之间的关系并不清楚。如果样本外 (Out of Sample) 变量的关系与样本内 (In Sample) 变量的关系完全不同，目前建立的回归方程 就不能正确地描述其关系。在样本外进行预测，当然就发生错误。

7.Logistic 回归

Logistic回归实际上是一种分类方法，主要用于二分类问题。 Logistic回归与多元线 性回归有很多相同之处，最大的区别是它们的因变量不同。两者可以归于同一个模型家 族，即广义线性回归模型 (Generalized Linear Model)。这一家族的模型形式类似，即样 本特征的线性组合，不同的是它们的因变量。如果因变量是连续的，就是多元线性回归， 如果因变量是二项分布，就是Logistic 回归。如果因变量是 Poisson 分布，就是Poisson 回归，如果因变量是负二项分布，就是负二项回归等。

为了了解 Logistic 回归，需要首先了解Logistic 函数(或称为Sigmoid 函数)。其函数

形式为  。这个函数的自变量的变化范围是(一∞,十∞),函数值的变化范 围是[0,1],函数的图像如图5- 18所示。



1 -

0.9——— —

0.8— — —— 0.7 — —--———

0.6—— —— 0.5—— —

0.4 ——--

0.3—— 0.2 —— 0.1 一

0

-6 0 6

图5-18 Sigmoid 函数图像

Logistic 回归分类器 (Logistic Regression Classifier) 的目的是从训练数据中学习出 一个0/1分类模型。这个模型以样本特征 (x₁,x₂,…,xn 是某样本数据的各个特征，维

度为n) 的线性组合0。+0Bx₁+…+O,zn= ∑0.x=θTx 作为自变量，使用Logistic函

数将自变量映射到(0,1)上。

上述线性组合代入Logistic 函数，构造一个预测函数

h₈ (x) 函数的值具有特殊的含义，它表示结果取1的概率。因此对于输入x, 分类结果为 类别1的概率为P(y=1|x;θ)=h(x), 分类结果为类别0的概率为 P(y=0|x;θ)=

**1-h(x)。**

现在有一个新的数据点Z, 新样本具有特征 z,Z₂, … ,Zn, 首先计算线性组合

θo+θ₁z₁+… 十+θnzn=θx, 然后代入 h₀ (x), 计算其函数值。如果函数值大于0.5,那 么类别y=1, 否则，类别y=0。 这里假设统计样本是均匀分布的，因此，设阈值为0.5。

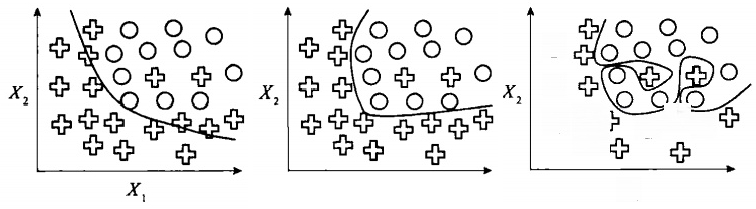
给定训练数据集，我们需要根据这些训练数据计算分类器的参数，也就是各个特征属 性**的加权参数θ=<θo,θ₁,θ2,…,θn>。具体的计算过程用到极大似然估计** **(Maximum**

Li**kelihood Estimation),可以使用梯度上升(下降)算法或者牛顿-拉菲森迭代算法求解。**

**Logistic 回归分类器适用于数值型数据和分类型数据，其计算代价不高，容易理解和** **实现，但是Logistic 回归分类器容易发生欠拟合** **(Underfitting) 现象，分类精度不高。** **图5-19显示了对平面上的两类样本点的回归分类器的效果，从左到右分别显示了欠拟** **合、拟合较好、过拟合的状况。**

Logistic回归分析可以应用于很多领域，下面举一个在流行病学中的应用。在流行病 学中，需要探索某疾病的危险因素，根据这些因素预测某疾病发生的概率。比如，要探讨 胃癌发生的危险因素，可以选择两组人群， 一组是胃癌组，一组是健康组。两组人群有不 同的体质特征和生活方式。对问题建模时，因变量为是否得胃癌，取值为“是”或者 “否”,自变量可以包括很多因素，比如年龄、性别、饮食习惯。自变量可以是连续的，也 可以是分类的。对采集的样本数据进行Logistic 回归分析，获得每个数据点(对应人)的 各个特征属性(对应上述各个自变量)的加权参数，就可以大致了解到底哪些因素是胃癌 的危险因素，即权重比较大的特征属性。当我们建立了这样的Logistic 回归模型，就可以 预测，在不同的自变量下发生某种疾病的概率有多大。

Logistic Regression Example



凸

凸

凸 凸 + + +

*X₁* X₁

(a) 欠拟合 (Underfitting) (b) 拟合较好 (Fitting well) (c) 过拟合 (Overfitting)

(a)h₉(x)=g(θo+θ₁x₁+θ₂x₂)(g=sigmoid function)

(b)hg(x)=g(θ₀+θ₁x₁+82x₂+0₃xi²+0₄xz²+θsx₁x₂)

(c)hg(x)=g(θo+θ₁x₁+θ₂x²+8₃x₁²x₂+θ4x₁²x₂²+0sx₁²x₂³+8₆x³x₂+…) **图5-19** **欠拟合与过拟合**

5.2.11 神经网络与深度学习 (Neural Network and Deep Learning)

人工神经网络是模仿动物和人类神经系统特征进行分布式并行信息处理的数学模型。 通过把大量人工神经元节点(感知机)连接起来形成神经网络，利用训练数据调整节点间 的连接强度，达到对新数据进行处理(包括分类、回归等预测能力)的目的。

人工神经网络技术可以追溯到20世纪40年代。1943年， Warren McCulloch 以及 Walter Pitts首次提出了神经元的数学模型。1958年，心理学家Frank Rosenblatt 提出了 感知器 (Perceptron) 的概念，在神经元的结构中加人了训练修正参数的机制，完成了人 工神经网络基本学理的构建。

1. 神经元(感知机)

神经网络由神经元(或者称为感知机， Perceptron) 组成。在计算机里对神经元进行 建模时，把它前端(模仿神经元的树突)收集到的输入信号进行加权求和，再通过一个激 活函数转换成输出，传送出去(模仿神经元的轴突)。图5-20展示了一个简单的神经元。

其中，x₁,x₂,x₃ 为神经元的输入，神经元的输出通过 )函数计算， f:R→R 称为激活函数。

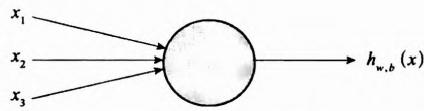


图5-20 一个神经元

一般来讲，激活函数为一个非线性函数，目的是对实际应用中输出和输入之间的非线 性关系进行建模。常用的激活函数有 Sigmoid 函数和 Tanh 函数，两个函数的具体形式 如下：



正是由于这个激活函数.神经网络具有对非线性关系进行建模的能力。

我们可以通过不断调整每个输入的权重，训练单个神经元。当输入一个样本数据后， 按照最小化输出误差 (Output Error) 的方向，来调整各个输入的权重。输出误差是实际 的输出值 (Actual Output) 和想要的目标值 (Desired Target)之间的差别。

一个神经元(感知机)能够完成简单的线性分类任务。人工智能专家Minsky 在1969 年出版了一本书Perceptron, 用数学方法证明了感知机的弱点，像异或 (XOR) 这样的简 单分类任务，都无法解决。

2. 带一个隐藏层的简单的神经网络

最简单的神经网络是前馈 (Feed Forward) 神经网络。在这个神经网络中，每一层的 节点仅和下一层的节点相连。除了前馈神经网络外，另外一种类型的神经网络称为递归神 经网络，它允许同一层节点相连或某一层的节点连到前面各层中的节点。在下文中，如果 不做特殊说明，我们讨论的将是前馈神经网络。

神经网络就是由许多单一的神经元连接而成的网络结构， 一个神经元的输出，可以是



另外一个神经元的输入。图5-21是一个简单的神经网络。该神经网络最左边的一层称为 输入层，最右边的一层称为输出层。中间的节点组成独立的一层，称为隐藏层，之所以称 为隐藏层，是因为我们不能从训练样本上观察到它们的取值。

|  |  |
| --- | --- |
| x₁  *x₂*  ₃  Layer1 | a“  a  a② Layer2 |

图5-21 一个简单的神经网络 (x₁,x₂,x₃

*h(x)*

a

Layer3

为输入层，不对输入数据做任何操作)

增加了一个隐藏层以后，神经网络系统不仅可以解决异或问题，而且具有非常好的非 线性分类效果。当网络中各层的节点数增大，神经网络权重优化的计算量较大，直到21 世纪初一直没有很好地解决。

在本文中，神经元有时也会用“单元”(Unit) 来指代，神经网络的表现形式是一个 有向图，因此有时也会使用“节点”(Node) 来表达同样的意思。在图5-21这个神经网 络里，输入层包含3个输入单元，隐藏层包含3个隐藏单元，输出层包含1个输出单元。 输入单元只负责传输数据，不做计算。

神经网络中上一层的各个神经元到下一层的各个神经元的连接有一个权重，这个权重 有一个初始值，经过训练(一般采用反向传播办法)获得一个优化的权重配置。当新的样 本值输入到训练后的神经网络，就可以获得一个输出值，这个输出值可以应用在对新的输 入数据进行分类和回归等目的。比如，在医疗诊断应用中，输入值可以是患者的各项生化 指标值，而不同的输出值就可以表示某种疾病是否存在。

3. 前向传播

计算输出值的过程称为前向传播。在前向传播过程中，每个神经元首先把上一层各个 神经元获得的数值进行加权(每个连接的权重)求和，然后应用激活函数，获得相应的输 出，再通过与下一层的连接传播给下一层的各个神经元，直到获得最后的输出。

以图5- 21为例， W,(1) 表示第l 层第j 单元与第l+1 层 第i 单元之间的连接参数，也 就是连接线上的权重，b:(1 表示第l 层 第i 单元的偏置项，也就是激活函数的常量部分。

a:(①表示第l层第i单元的激活值(输出值),当l=1 时 ，a;(1)=xi。

激活过程可以用如下公式表示出来：

a₁(2)=f(w₁(1)x₁+w12(1)x₂+w13(1)x₃+b₁(1))

a₂(²)=f(w₂1(1)x₁+w₂2(1)x₂+w₂3(1)x₃+b₂(1)

a₃(²)=f(w₃1(1)x₁+w₂(1)x₂+w₃3(1)x₃+b₃(¹)

hw.b(x)=a₁(3)=f(w₁(2)a₁(2)+w₁2(2)a₂(²)+w₁₃(²)a₃(²)+b₁(2')

在实际应用中，我们可以创建包含多个隐藏层的神经网络。在深度学习技术崛起之 前，一个隐藏层的神经网络在很多应用中已经足够。理论证明，包含1个输入层/1个隐藏 层/1个输出层的神经网络可以无限逼近任意连续函数，对数据中表现的非线性关系进行 建模。

至此，我们看到神经网络的本质，就是通过参数与激活函数来拟合特征与目标之间的 真实函数关系。



4. 反向传播 (Back Propagation) 训练算法

神经网络的权重使用一种称为反向传播 (Backpropagation.BP) 的方法进行训练。 该算法于1986年由 Rumelhar 和 Hinton 等人提出，它解决了带隐藏层的神经网络优化的 计算量问题，使得带隐藏层的神经网络走向真正的实用。

形象地描述反向传播算法，开始在输人层输入特征向量.经过神经网络各个隐藏层的 层层计算获得输出，输出层发现输出和正确的输出(训练数据的输出部分)不一样，这时 它就让最后一层神经元进行参数调整。最后一层神经元不仅自己调整参数，还会要求连接 它的倒数第二层神经元调整连接权重，并且逐层往回退，调整各个神经网络层间连接的 权重。

具体来讲，假设有一个样本集 (x(i,yi),x(i 为多维向量，y()可以为多维(比如2 维)或者一维向量。反向传播算法分两步，即正向传播和反向传播。正向传播即使用x(i) 作为输入，输入的样本从输入层，经过隐藏层， 一层一层处理以后，由输出层进行输出。 在逐层处理的过程中，每一层神经元的状态只对下一层神经元的状态产生影响。

在输出层上获得一个输出以后，和期望的输出值y(进行比对，发现不相等，进入反 向传播过程。反向传播过程把误差信号按照原来正向传播的通路的相反方向传回，并且对 每个隐藏层的各个神经元的连接权重进行修改，目的是使得误差信号趋向最小。

BP 算法的本质①是误差函数的最小化问题。误差函数的定义一般采用期望输出和实 际输出的差的平方和，即



式中，x,'为第l 层的实际输出；y” 则为期望的输出.也就是训练样本里对应xi 的输出。 一般采用非线性规划中的梯度下降方法来修改权重。对于某个神经元的某个权重的更

新，采用  的公式。式中，E 为输出误差；w, 为输入到该神经元的第i 个连接 的权重；a 为学习率。由于误差函数是非线性的，梯度下降的方法可能会陷人局部最小值。

神经网络的训练过程试图优化出模型的参数，使得其在训练数据集上获得较小的误 差。我们使用这个模型，最终是要对真实场景的新数据进行分类和预测。于是，我们把已 有的数据集分割成训练数据集和测试数据集，在测试数据集上验证模型的预测性能，以此 了解模型的泛化能力 (Generalization, 即处理新数据的能力)。

神经网络模仿了动物和人类神经系统的行为特性，经过训练的神经网络能够对非线性 关系进行建模，在分类和预测方面获得了较好的性能。1986年提出的反向传播算法，解 决了训练效率问题，神经网络被应用到语音识别、图像识别、自动驾驶等多个领域，也获 得了较好效果。

但是上述神经网络存在若干问题。首先，尽管使用了 BP 算法，一次神经网络的训练 仍然耗时太久，而且训练过程可能导致局部最优解，这使得神经网络的优化较为困难。此 外，隐藏层的节点数需要根据应用调整，节点数设置的多少，会影响到整个模型的效果， 给实际应用带来不便。

20世纪90年代中期，Vapnik 等人发明了SVM(Support Vector Machine,支持向量

①<https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>给出了一个反向传播算法的实 例，<http://www.emergentmind.com/neural-network> 则给出了反向传播算法的可视化效果，读者可以参考。

机)技术。 SVM 在若干方面体现出比神经网络更大的优势，比如无须调整参数，训练和 执行效率高，可以获得全局最优解等。SVM 技术在20世纪90年代到21世纪初迅速代替 神经网络，成为更流行的机器学习算法，直到深度学习技术的崛起。

**5.** **深度学习**

深度学习是21世纪初流行起来的机器学习方法，它依赖于更深层次(包含多个隐藏 层)的神经网络。深度学习在图像识别、语音识别、自然语言处理、机器人等领域获得了 超过传统机器学习方法的性能。在人脸识别 (Face Recognition) 比赛LFW 和自然图像分 类比赛 ImageNet 中，获得了超过人类的识别能力。2016年Google 的 Alpha Go 围棋程序 击败了人类棋手李世石九段，2017年经过改进的Alpha Go具有更强的棋力，打败了世界 排名第一的柯洁九段，显示了深度学习技术的强大威力。

深度学习能够流行起来的原因包括几个方面：大数据集的积累、计算机运算能力的提 高、深度学习训练算法的改进，以及深度学习模型具有能够自主从数据中学习到有用的特 征 (Feature) 的特点等。

大数据是深度学习的原材料。如果没有大数据，更加复杂的神经网络也无法得到很好 的训练，人们无法获得好的模型。在深度学习的实际应用方面取得突破性进展的，大多是 拥有大数据的互联网公司，比如 Google,Facebook,Microsoft,Baidu 等。

硬件的进步是深度学习流行起来的第二个原因。 GPU 性能的提高以及超级计算机和 云计算技术的迅猛发展，使得深度学习的实现具有了硬件基础。其中，高性能图形处理器 极大地提高了数值和矩阵运算的速度，使得机器学习算法的运行时间得到了显著的改善。 2011年.Google Brain团队(机器学习专家Andrew Ng 和分布式系统专家Jeff Dean 领 导)用1000台机器、16000个CPU 实现的深度学习模型，包含10亿个连接(1 Billion Connections) 。 使用来自YouTube 的大量视频进行训练以后，该神经网络能够自动识别出 猫脸。①到了2015、2016年，这样的模型可以在少量高性能的GPU 上实现。深度神经网 络的训练过程完全可以并行处理，使用GPU 大幅提升了其训练的速度。

更深层次的神经网络，如果没有有效的训练方法，其训练过程是很慢的，根本没有办 法实用。深度学习流行起来的另外一个因素是人们找到了提高深度神经网络模型训练效率 的方法。主要的贡献者是多伦多大学的 Hinton 教授。2006年，他在 Science 杂志上发表 了深度学习的里程碑式的论文，将深度学习的训练效率提升了一大截。深度神经网络在训 练上的难度，可以通过“逐层预训练”(Layer-wise Pre-training) 来有效克服，Hinton 在 论文中给出了无监督的逐层预训练方法。除了Geoffrey Hinton, 一批学者在深度学习方 面也做出了重大贡献，包括 Yoshua Bengio,Yann LeCun,Andrew Ng 等。

除了训练性能提高之外，深度神经网络可以自动识别样本的特征。这一点使得深度学 习在一些不知如何设计有效的特征的应用场合，比如图像识别和语音识别等，获得了很好 的性能。在神经网络中，浅层的神经元学习到初级的简单的特征 (Primitive Feature),馈 人下一层神经网络层。深层的神经元在前一层神经元识别到的特征的基础上，学习到更加 复杂的特征 (Complex Features)。这个过程在相邻的神经网络层间重复，各个神经网络层 学习到不同抽象级别的特征，越是靠后的神经网络层，学习到更加抽象的特征，最后完成 预定的识别任务，比如语音识别和图像识别。

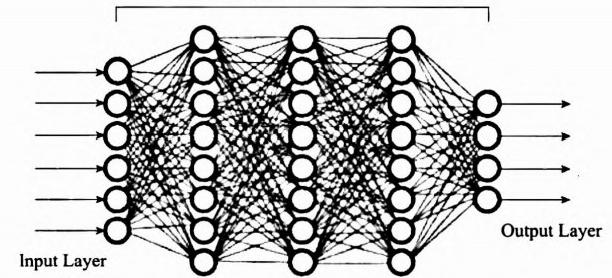
比如，在图像识别中，第一个隐藏层学习到的是“边缘”的特征，第二个隐藏层学习

①<https://googleblog.blogspot.com/2012/06/using-large-scale-brain-simulations-for.html.>

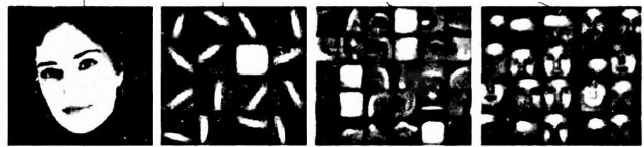
到的是由“边缘”组成的“形状”的特征，第三个隐藏层学习到的是由“形状”组成的 “图案”的特征，最后的隐藏层学习到的是由“图案”组成的“对象”的特征等。纽约大 学的Zeiler 和 Fergus 对这个问题进行了验证，他们把深度神经网络中某些神经元挑选出 来，把在它们上获得较大响应的输入图像放在一起，观察其共同特点，他们发现中间不同 隐藏层的神经元，分别响应了不同抽象级别的图像特征。这些不同的抽象层次，是网络中 间各个隐藏层神经元的偏好，它们的输出又作为后续隐藏层的输入，形成由低到高的不同 抽象级别的特征，于是更高层次的概念从低层次的概念学习得到。

图5-22显示了一个深度神经网络实例及其各个隐藏层对图像的不同抽象级别的特征 的识别能力，第一个隐藏层学习到(识别出)边缘，第二个隐藏层学习到(识别出)边缘 的组合(各种形状),第三个隐藏层学习到(识别出)人脸，即各个隐藏层依次学习到 (识别出)越来越复杂的人脸特征。

Deep Neural Network



Hiden Layer 1 Hidden Layer 2 Hiden Layer 3



edges combinations of edges objeet models

**图** **5** **-** **2** **2** **深度神经网络的隐藏层及其特征抽象能力**

资料来源：<https://www.saagie.com/fr/blog/object-detection-part1.>

表5-16总结了语音识别、图像识别、自然语言理解等任务中特征所具有的天然的层 次结构 。

**表5-16** **几种任务领域的特征层次结构**

|  |  |
| --- | --- |
| 任务领域 | 原始数据→浅层特征→ 中层特征→高层特征→训练目标 |
| 语音识别 | 样本 频段 声音 音调 音素 单词 语音识别 |
| 图像识别 | 像素 线条 纹理 图案 局部 物体 图像识别 |
| 自然语言理解 | 字幕 单词 词组 短语 句子 段落 文章 语义理解 |

隐藏层是神经网络对训练数据进行内部抽象表示 (Internal Abstract Representation) 的结构，就像人脑对现实世界的对象有一个内部表示一样。在神经网络里增加隐藏层，使

**得后续的隐藏层可以在前导隐藏层的内部表示的基础上，建立新的抽象级别的内部表示。**

**可以说，深度模型是技术手段，特征学习是目的。与传统的浅层学习对比，深度学习**

的不同之处在于：(1)强调模型结构的深度，深度神经网络通常有5层、6层，甚至超过

**10层的隐藏层；(2)突出特征学习的重要性，通过逐层的特征变换，将不同抽象级别的**

特征识别出来，最后使得分类和预测更加容易。

正是借助于算法的改进、计算平台性能的提高，人们可以模拟更多的神经元组成的复杂 神经网络的行为。万事俱备，东风也来了，大数据成为复杂人工神经网络训练的原材料。

2012年多伦多大学的Krizhevsky 等构造了一个大型的卷积神经网络，该网络共有9 层，65万个神经元，6000万个参数。网络的输入是图片，输出是1000个图片分类，表 示不同的对象类别，比如美洲豹、救生艇等。他们使用大量的图片训练这个模型，最后在 ImageNet 图片分类方面，识别性能优于当时所有的其他分类器，错误率由25%降低为 17% 。ImageNet 是斯坦福大学Li Feifei 教授创建的到目前为止最大的图像识别 (Image Recognition) 数据库，共包含大约22000个类，15M 左右的标注图像。其中，目前最常 用的 LSVRC-2010 Contest数据集，包含1000个类，1.2M 图像。表5-17总结了浅层神 经网络模型和深层神经网络模型的区别。

**表5-17** **浅层神经网络vs.深层神经网络**

|  |  |  |
| --- | --- | --- |
|  | 浅层网络 | 深层网络 |
| 层数 | 1～2层 | 5～10层，甚至更多 |
| 特征提取方式 | 特征工程 | 自动抽取特征 |
| 模型特点 | 凸代价函数  可以收敛到全局最优 | 非凸的代价函数  存在大量的局部最优点 容易收敛到局部最优 |
| 模型表达能力 | 有限 | 强大 |
| 训练难度 | 容易 | 困难 |
| 应用领域 | 时间序列分析、故障诊断 …… | 语音识别、图像识别、自然语言理解…… |

**6.** **深度学习的应用**

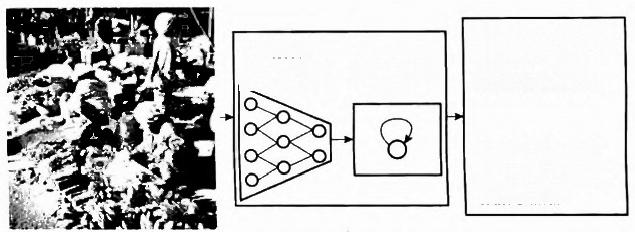
深度学习的应用非常广泛，包括图像/视频的识别、语音识别、自然语言处理等。

在图像和视频应用方面，深度学习模型可以识别照片中的物体，对照片进行自动分类和 搜索，比如Google Photo、百度识图、淘宝拍立淘等，都使用了深度学习模型。深度学习模 型应用于自动驾驶系统，对人员、车辆等路况信息进行识别和追踪，进而做出有效的应对。 深度学习模型还可以用于人脸识别，实现刷脸支付等功能，为人们的生活带来方便。

2014年Google 试图利用深度学习技术，从图像直接生成一段自然语言的描述。 Google 把两个深度神经网络结合起来(卷积神经网络CNN 和循环神经网络 RNN) 组成 一个模型，完成这个任务。其中一个神经网络负责图像识别，另一个神经网络负责语言生 成。图5-23展示了他们使用的网络结构。这个网络结构，基于左边的图片，生成了右边 的文字描述。

2014年香港中文大学教授汤晓鸥、王晓刚及其研究团队，利用深度学习技术研发的 DeepID 人脸识别技术，准确率达到99.15%,比人用肉眼识别更加精准。 LFW 是人脸识 别领域使用最广泛的测试基准。人用肉眼在 LFW 上的识别率为97.52%。此前，该研究 组早已在2011年开始深度学习方法的研究，在2013年把 LFW 上的识别率提高 到92.52%。



A group of people shopping at an

Language

Generating

RNN

Vision

Deep CNN

outdoor market.

There are many vegetables at the fruit stand.

**图5-23** **自动生成图片的描述**

2015年末微软亚洲研究院利用深度达152层的“深层残差网络”,参加ImageNet 挑 战赛。该网络的层数，比以往任何成功使用的神经网络的层数多5倍以上，并且有效避免 了梯度消失。他们以绝对优势获得图像分类、图像定位以及图像检测全部三个主要项目的 冠军。他们在另一项图像识别挑战赛 MSCOCO(Microsoft Common Objects in Context, 常见物体图像识别)中同样获得冠军。2014年 ImageNet 挑战赛获胜的系统错误率为 6 . 6%,2015年微软系统的错误率已经低至3 . 57%。早先，该团队在2015年1月首次实 现了超过人类视觉分类能力的突破，他们的系统的错误率已降低至4.94%,在同样的实验 中，人眼辨识的错误率大概为5 . 1%。

在语音识别方面，深度学习也得到了广泛应用。在深度学习技术的帮助下，计算机拥 有强大的语音识别 (Speech Recognition) 能力，人机交互的模式将变得更加丰富。 Google(Now)、Apple(Siri)、Microsoft(Cortana)、Baidu(Deep Speech)、科大讯飞 等公司都推出了其语音识别产品。2012年12月，微软亚洲研究院展示了中英即时传译系 统，集成了语音识别、机器翻译和语音合成技术，其错误率仅为7%。①2012年Google 公 司利用深度学习技术，大幅度改善了Android 操作系统上的语音识别的精度。使用深度学 习技术，语音识别的成功率大大提高，尤其是在嘈杂的环境中，智能手机语音识别系统的 错误率下降到了25%,语音搜索结果也有了不小的改善。2013年多伦多大学的 Alex Graves,Abdel-rahman Mohamed以及Geoffrey E.Hinton 等，使用双向LSTM/RNN 网 络，打破了著名的TIMIT 语音识别测试的纪录。TIMIT 语音库为研究中常用的语音库， 适用于语音识别、说话人识别等语音信号处理。

在自然语言处理方面，深度学习也大有用武之地。其中，Google 公司于2016发布了 在机器翻译方面的最新成果。他们使用基于LSTM 神经网络的深度学习模型，比目前的 基于统计方法的机器翻译引擎获得了更好的翻译结果。他们的系统对于长句子的机器翻 译，也获得了更高的准确度。

深度学习让计算机拥有图像、视频、语音的识别能力。深度学习技术也在改变着机器 人领域，帮助机器人更好地感知周围的世界。除此之外，深度学习技术还应用到互联网搜 索、广告推荐、量化交易、医疗大数据分析等众多领域。实际上，凡是需要对大数据进行 分析、预测未知信息的领域，深度学习都可以派上用场。

需要注意的是，深度学习并非万能的技术，人们不应对其过于迷信，它学习到的可 能是数据中的相关关系，但不一 定是因果关系。深度学习依赖于大数据。对于人类来讲 轻松不过的事情，机器就要看成千上万张图片才能总结出来。比如苹果究竟有什么样的特



①<http://tech2ipo.com/56452.>



征，然后按照总结出来的特征识别下一个苹果。如果机器能够通过小数据学习，才是更像 人类的学习方式。“比如一个小孩，一次他看见一个苹果以后，下次再看见，他就能认识 这是个苹果，不需要看成千上万个苹果”(张亚勤)。在小数据学习方面，理论上目前并没 有实质性突破。

此外，深度学习的能力虽然很强，但是和我们预期的真正的强人工智能相比，仍然缺 乏必要的能力，比如逻辑推理的能力，集成抽象知识的能力。因此，深度学习技术可以看 作实现人工智能的一种途径，而不是终极解决方案。

在工程实践中，把深度学习技术和其他机器学习技术(比如贝叶斯推理和演绎推理等 技术)结合起来，互相取长补短，是一个有前途的策略。此外，深度学习技术可以和增强 学习 (Reinforcement Learning) 技术相结合。增强学习是指计算机通过与环境交互，从 中得到奖赏和惩罚，进而自主学习 (Self-Learning) 的策略。

2016年3月击败李世石的围棋程序Alpha Go,使用了深度学习、增强学习、蒙特卡 洛树搜索 (Monte Carlo Tree Search,MCTS) 等方法，验证了深度学习技术和其他机器 学习技术结合，可以使计算机能够不断自主学习，从而获得高度优化的训练模型。

从长远来看，目前美国和欧盟以及中国发起的脑科学计划，将使人们对大脑的神经活 动有更加深人的了解。利用这些知识改善深度学习模型的建模，可能是深度学习取得长足 发展的契机。①

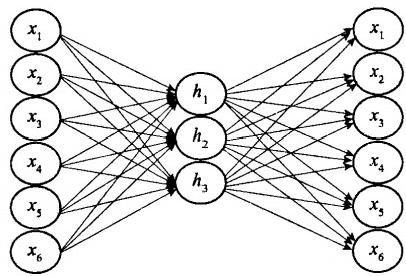
**7.** **深度神经网络的基础模块和网络模型**

接下来将介绍深度神经网络的基本模块，包括 Autoencoder.Restricted Boltzmann Machine(RBM) 等。然后介绍不同的神经网络模型，包括 CNN.DBN,RNN,

LSTM 等。

(1)Autoencoder (自动编码器)。

通常来讲，一个Autoencoder 就是一个前向反馈的简单的神经网络，它的目的是从训 练数据上学习一个经过压缩的简洁的数据的表示 (Compact Representation) 或者简洁的 一个编码(见图5-24)。为了达到这个目的，输入层和输出层的节点数是一样的，预期的 输出就是训练数据本身，隐藏层的节点数大大少于输入层(或者输出层的节点数)。



**图5-24** **一个** **Autoencoder**

这个网络的目的不是让其学习到输入数据和预期结果(分类标签或者预测值)之间的



①Jack Clark. 深度学习“教父”: AI 应从神经科学中借鉴想法和思路 .<http://www.cf.org.cn/c/2017-05-23/>

595978.shtml.

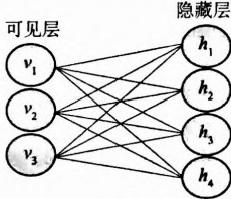
一个映射，而是学习到一种数据的特征或者内部表示的结构 (Internal Representation Structure)。于是，隐藏层也叫做特征检测器 (Feature Detector)。

隐藏层的单元数量大大少于输入层或者输出层的单元数量，迫使网络学习到的是数据 呈现的最重要的特征，从而达到降维 (Dimensionality Reduction) 的目的。

(2)Restricted Boltzmann Machines(RBM, 受限玻尔兹曼机)。

标准的玻尔兹曼机由可见单元层 (Visible Layer, 扮演输入/输出层)以及隐藏层 (Hidden Layer) 构成。可见单元层和隐藏层的单元之间是双向全联通的，也就是数据可 以从可见层的单元传播到隐藏层的单元，也可以从隐藏层的单元传播到可见单元层的单 元；可见单元层的每个单元和隐藏层的每个单元之间都有连接。如果可见单元层和隐藏层 的单元不是全联通的 (Fully Connected), 这样的玻尔兹曼机称为受限玻尔兹曼机。 一般 来讲，各个单元的激活函数产生0或者1,这些0/1输出符合伯努力分布 (Bernoulli Dis- tribution)。

RBM 是一种生成式随机神经网络 (Generative Stochastic Neural Network), 它可以 从输入的数据集学习到其概率分布(见图5-25)。



**图5-25** **一个** **RBM**

RBM 采用一种称为对比分歧 (Contrastive Divergence,CD) 的方法进行训练。 CD 算法的一次迭代过程包括两个阶段。

Positive 阶段：样本数据v 输入神经网络的输入层。v 通过和前向反馈网络类似的前向 传播方法，传播到隐藏层，隐藏层的激活结果为h。

Negative 阶段：把h 传播回可见层(注意上文中提到，隐藏层和可见层的单元连接是 双向的),结果为v', 把 v '传播到隐藏层，激活结果为h′。

在此基础上，对神经网络的权重进行更新，具体为：

*w(t+1)=w(t)+a(vh-v'h'T)*

式中，α为学习率；v,v',h,h′,w 为向量。

这个算法背后的思想是，Positive 阶段(由v 产生h), 使得网络获得了实际数据的内 部表示 (Internal Representation) 。Negative 阶段，则尝试利用内部表示，重新创建真实 的数据(由h 产生v', 然后h′) 。权重更新的公式的目的是使得产生的数据尽可能和原来的 实际数据 (Real World Data) 接近，误差尽可能小。对比分歧方法提供了一种对最大似然 的近似，在这里用于学习受限玻尔兹曼机的权重。

这个算法使得神经网络学习到对输人数据如何进行内部表示，然后用这些内部表示重 新生成数据。如果重新生成的数据和实际数据还未达到足够接近，那么该网络进行权重的 调整，然后接着尝试。

经过若干次迭代之后(几百次),我们就能够观察到，部分隐藏层单元对某些类别的



**数据产生更高的激活值，其他部分隐藏层单元则对另外类别的数据产生更高的激活值。** **RBM已经学习到了数据的某种内部表示。**

**(3)堆叠的自动编码器** **(Stacked Autoencoder)。**

**Autoencoder 可以堆叠起来，构成一个深度网络。这样的深度网络可以每次一层地进** 行**训练，解决传统反向传播** **(BP)** **算法带来的梯度消失** **(Vanishing** **Gradient)** **问题和过** **拟合问题** **(Overfitting)。**

**梯度消失问题是指当我们在神经网络中加入更多的隐藏层时，反向传播过程很难把修** **正信息传播回前面的隐藏层，本应用于修正模型参数的误差随着层数的增加而呈指数递** **减，相对于各个隐藏层之间的连接的权重来讲，开始变得很小，这导致了模型训练的效率** **低下，无法获得好的训练模型。**

**过拟合则是训练出来的模型对训练数据拟合过度，但是当用这个模型来对新的数据进** **行分类和预测时，模型表现不是很好，也就是缺乏泛化能力。近年来出现了新的实践上非** **常有效的正则化方法** **(Regularization), 提高模型泛化能力，比如** **Dropout 和** **Drop Con- nect, 以及数据扩增** **(Data Augmentation) 技术等。**

**把** **Autoencoder 堆叠起来的深度网络具有更强大的分类和预测能力，获得令人印象深** **刻的结果。上文提到过的Google著名的识别猫脸的论文，就是使用把Autoencoder 堆叠** **起来的深度网络，对无标签数据进行学习实现的，显示了深度网络的无监督学习能力。**

**在堆叠的Autoencoder 构成的深度网络里，隐藏层** **(Autoencoder)t 的输出，作为隐** **藏层t+1 的输入，第一个隐藏层的输入就是整个网络的输入，也就是训练数据集的输入** **数据** **。**

**堆叠的** **Autoencoder,** **使用逐层的贪心训练算法来进行训练，具体过程如下**：

①用输入层、隐藏层、输出层构造一个网络，使用所有的训练数据，使用反向传播算 法，训练第一个堆叠的 Autoencoder。注意上文中介绍Autoencoder 时已经提到的，输出 **数据和输入数据是一样的。**

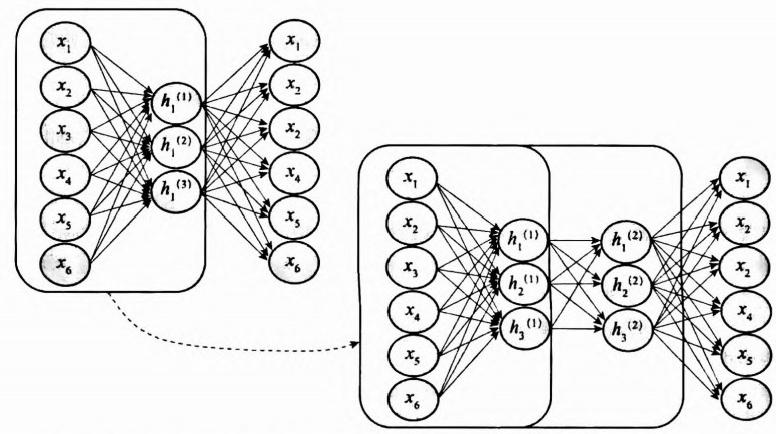
②把第一个堆叠的Autoencoder的输出层剥离，把第一个堆叠的Autoencoder的隐藏 层作为第二个堆叠的Autoencoder 的输入，再附加一个输出层，训练第二个堆叠的 Au- toencoder。训练数据输入到第一个堆叠的Autoencoder, 第一个堆叠的Autoencoder 的隐 藏层的输出传播到第二个堆叠的 Autoencoder 的隐藏层，第二个堆叠的Autoencoder 的隐 藏层的连接权重，通过反向传播算法进行更新，如图5-26所示。

③对所有堆叠的Autoencoder 采用上述办法进行训练，也就是把上一个堆叠的 Au- toencoder的输出层剥离，然后堆叠新的Autoencoder, 即加入一个隐藏层和输出层，用反 **向传播算法进行训练。**

④步骤1～3称为预训练过程 (Pre-Training), 使得网络的各层单元之间的连接的权 重初始化。经过预训练的网络并没有获得输入数据到输出的分类标签的映射模型。

逐层初始化完成后，需要增加一层(或者几层)全连接的输出层，对应神经网络应该 输出的分类标签或者预测值，然后用有标签的数据对整个网络用反向传播算法进行整体有 监督的训练，这个过程称为精细调优 (Fine Tuning)。可以看出，预训练过程初始化了网 络的各个层次连接的权重，从而让我们获得了一个复杂的多层的神经网络，再经过进一步 的训练 (Fine Tuning) 就可以获得最后的模型。由于深层神经网络模型有很多的局部最 优解，模型初始化的位置将决定最终模型的质量，预训练过程让模型处于一个较为接近全 局最优的位置，精细调优过程则想办法把模型调整到全局最优。





**图5-26** **从第一个堆叠的Autoencoder的训练到第二个堆叠的Autoencoder的训练**

分层预训练相当于对输入数据进行逐级抽象，类似于大脑的认知过程。1958年， Da- vid Hubel 和 Torsten Wiesel 发现了一种称为“方向选择性细胞”(Orientation Selective Cell) 的神经元细胞。当瞳孔发现了眼前的物体的边缘，而且这个边缘指向某个方向时， 这种神经元细胞就会活跃。

这个发现激发研究人员对神经网络进一步思考。从神经末梢到神经中枢到大脑的工作 过程，或许是一个不断迭代、逐级抽象的过程，即从原始信号.做低级抽象，逐级向高级 抽象迭代， 一直到更高级别的抽象的概念，比如眼睛的神经末梢受到刺激(瞳孔摄入像 素),神经细胞经过初步处理，发现边缘和方向，另一部分神经细胞进一步抽象，判断物 体的形状，然后其他神经细胞进一步抽象，大脑判断出具体的物体类别。人类的逻辑思维 则使用高度抽象的概念。

(4)深度信念网络 (Deep Belief Network,DBN)。

就像堆叠Autoencoder 一样，我们也可以堆叠RBM 。堆叠RBM 构成的网络称为深度 信念网络。在这种情况下，第t 个堆叠的RBM 的隐藏层作为第t+1 个堆叠的RBM 的可 见层。第一个堆叠的RBM 的输入层就是整个网络的输入层。

对深度信念网络，使用逐层贪心预训练算法进行训练。预训练过程如下：

①使用所有的训练数据，对第一个堆叠的RBM 使 用CD 算法进行训练。

②把第一个堆叠的RBM 的隐藏层作为第二个堆叠的RBM 的可见层。把训练样本输 入到第一个堆叠的RBM 的可见层，输入数据通过第一个堆叠RBM 的层间连接权重传播 到其隐藏层。第一个堆叠的RBM 的隐藏层，作为第二个堆叠的RBM 的输入，使用CD 算 法训练第二个RBM。

③重复步骤①~②,对所有堆叠的RBM 进行训练。

④和堆叠的 Autoencoder 类似，经过预训练过程，我们对整个网络进行扩展，增加一 层输出层，这个输出层全连接到最后一个堆叠的RBM 的隐藏层。

简而言之，从上述过程我们看到当单层RBM 被训练完毕后，另一层 RBM 可堆叠在 已经训练完成的RBM 上，形成一个多层模型。每次堆叠时，训练样本输入原有的多层网

络输入层，权重则是先前训练得到的权重，该网络的输出作为新增 RBM 的输入，新的

**RBM** **重复先前的单层训练过程。**

经过预训练，我们得到一个层间连接的权重得到初始化的深度网络，接着可以使用反 向传播算法进行精细调优 (Fine Tuning)。从上述描述我们可以看到，整个深度网络的结 构类似于堆叠的Autoencoder, 只不过每个Autoencoder 被替换成RBM, 逐层训练的算法 由反向传播替换成了CD 算法而已。

(5)卷积神经网络 (Convolution Neural Network,CNN)。

卷积神经网络是一种特殊类型的前向反馈神经网络，特别适合于图像识别、语音分析 等应用领域。卷积神经网络由于一个映射面上的神经元共享权值，因而减少了网络自由参 数的个数，降低了网络参数选择的复杂度。这个优点在网络的输入是图像时表现得更为明 显，可以使用图像直接作为网络的输入，避免了传统识别算法中复杂的特征提取过程，并 且在一个映射面共享权值，使得图像的特征被检测出来而不管它的位置 (Location) 是否 发生移动。此外，卷积神经网络是识别二维形状而特殊设计的一个多层网络结构，这种网 络结构对平移、比例缩放、倾斜或者其他形式的变形，具有高度不变性。

为了深入了解卷积神经网络，我们首先了解卷积操作。卷积操作就是一个图像过滤器 (Image Filter), 它定义为对一个矩形图像区域进行加权求和操作。比如，我们要对图像A 进行卷积操作，产生图像 B, 卷积操作的过滤器为6×6的权重矩阵。那么图像 B 的 < 1 , 1>位置的像素的值，是A 图像从<1,1>像素开始的矩形区域和6×6的权重矩阵的加权 和 (Weighted Sum), 图像B的<1,2>位置上的像素的值，是A 图像<1,2>像素开始 的矩形区域和6×6的权重矩阵的加权和等。

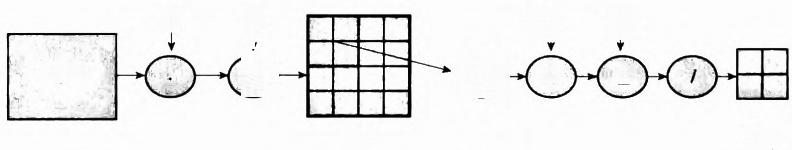
卷积神经网络是一个多层的神经网络结构，各层为C 层 (Convolutional, 卷积)或者 S 层 (Subsampling, 子采样)。 一般地，C 层为特征提取层，S 层是特征映射层，C 层和S 层交替组织起来。特征映射结构采用Sigmoid 函数作为激活函数，使得特征映射具有位移 不变性.这点有利于图像的识别。2011年以来，前馈神经网络深度学习中的常用方法是 交替使用卷积层 (Convolutional Layer) 和最大值池化层 (Max Pooling Layer, 即子采样 层),并加入单纯的分类层(即输出层)而构建。

卷积层一般对输入使用一个或者多个过滤器 (Filter) 进行转换，把这个过滤器应用 到输入上(图像),得到的结果称为特征图 (Feature Map)。卷积神经网络使用多个卷积 核，形成对输入的多个特征图 (Feature Map)。如果前后两个层次都是卷积层，那么每个 输出FM(Feature Map) 连接到每个输入FM 。多个过滤器的使用使得卷积神经网络利用 它们，分别检测数据(图像)中的不同特征集合 (Feature Set)。

S 层用于对输入(图像)进行子采样。比如输入是32×32像素的图像，如果子采样 的区域是2×2,那么输出将是16×16的图像，也就是说每4个(2×2)原图像的像素， 被合并到输出图像的一个像素。子采样的方法很多， 一般采用最大 (Max) 池化、平均 (Average) 池化和随机 (Stochastic) 池化等。

图5-27展示了卷积和子采样的过程。其中，卷积过程包括用一个可训练的滤波器 f, 去卷积一个输入的图像(第一阶段是输人的图像，后面的阶段就是卷积特征图了), 然后加一个偏置b, 得到卷积层 C。 卷积运算可以看作从一个平面到另一个平面的 映 射 。





b

∑

S

② X

*f,*

**Input**

b

C.

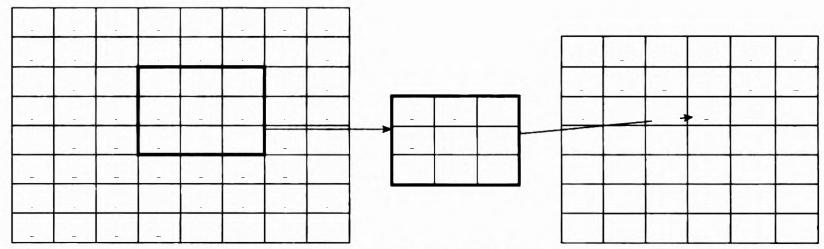
.

**图5-27** **卷积和子采样的过程**

子采样过程包括，每2×2=4个像素构成的区域求和变为一个像素，然后通过标量 Wx+1加权，再增加偏置bx+1, 然后通过一个Sigmoid 激活函数，产生一个缩小4倍的特征 映射图Sz+1。S 层可以看作模糊滤波器，它起到二次特征提取的作用。

为了对卷积操作有一个更深的认识，图5-28展示了对一幅8×8分辨率的图像使用 卷积方法提取特征的过程，在这里使用了3×3的卷积核，处理完原图像的每个3×3分辨 率的区域之后，最终得到6×6分辨率的输出图像。

卷积神经网络的最后一个C 层或者S 层，通过全连接方式连接到输出层，输出层输出 分类标签或者预测值。



0

1

1

工

0

1

0

工

工

1

1

0

1

0

1

0

0

0

1

1

1

1

1

0

0

1

0

0

0

1

0

1

1

1

0

1

0

0

0

1

1

0

1

0

0

0

1

0

0

0

0

1

1

1

0

0

0

4

3

5

3

5

1

1

0

1

0

1

0

1

4

3

5

1

4

0

1

0

1

0

1

1

0

1

0

2

2

1

工

Image 8×8 Conv filter 3×3 Convolved feature 6×6

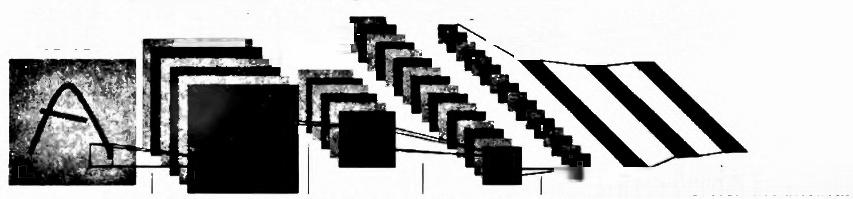
**图** **5** **-** **2** **8** **卷积操作过程**

(6)卷积神经网络实例。

用于手写体(Digit Recognition)识别的LeNet-5 是一个卷积神经网络(见图5-29), 不算输人层，它一共有7层，每层都包含可以训练的连接权重。

C3:f.maps 16@10×10

C1:feature maps S4:f.maps 16@5×5



C5:layerF6:layerOUTPUT 120 84 10

Full connectionl Gaussian connections

S2:f.maps 6@14×14

6@28×28

Z 器

INPUT 32×32

Convolutions Subsampling Convolutions Subsampling Full connection

**图** **5** **-** **2** **9** **卷积神经网络实例**

资料来源：<http://yann.lecun.com/exdb/lenet/index.html.>

①输入为32×32大小的图像。

②C1 层是一个卷积层，使用6个滤波器。 C1 层由6个特征图 (Feature Map) 构成，

特征图中每个神经元与输入中5×5的邻域相连，特征图的大小为(32-5+1)×(32- 5+1)=28×28。由于相同Feature Map 共享权值，不同Feature Map 之间权值不同。每 个滤波器5×5=25个Unit 参数和1个Bias 参数， 一共6个滤波器，共(5×5+1)×6= 156个可训练参数，共(5×5+1)×6×28×28=122304个连接。

③S2 层是一个子采样层，对图像进行子抽样，可以在减少数据量的同时保留有用信 息。S2 层有6个14×14的特征图。特征图中的每个单元与C1 中相对应特征图的2×2邻 域相连接。S2 层每个单元的4个输人相加，乘以1个可训练参数，再加上1个可训练偏置， 通过Sigmoid 函数计算结果。S2 层有(1+1)×6(特征图)=12个可训练参数。对于S2 层 的每一个图的每一个点，连接数是2×2+1(偏置),总共是(2×2+1)×14×14×6= 5880个连接。

可训练系数和偏置控制着 Sigmoid 函数的非线性程度。如果系数比较小，那么运 算近似于线性运算，子采样相当于模糊图像。如果系数比较大，根据偏置的大小，子 采样可以被看成是有噪声的“或”运算，或者有噪声的“与”运算，每个单元的2× 2感受视野并不重叠，因此 S2 中每个特征图的大小是 C1 中特征图大小的1/4(行、 列各1/2)。

④C3层也是一个卷积层，它同样通过5×5的卷积核去卷积层 S2, 得到的Feature Map 就只有10×10个神经元。它有16种不同的卷积核，于是存在16个 Feature Map。

需要注意的是，C3 中的每个Feature Map是连接到S2 中的所有6个或者某几个Fea- ture Map的，表示本层的Feature Map是上一层提取到的Feature Map的不同组合。从 C3的角度看，它有16个图。C3 的前6个特征图以 S2 中3个相邻的特征图子集为输入， 接下来6个特征图以S2 中4个相邻特征图子集为输入，接着的3个特征图以不相邻的4 个特征图子集为输入，最后一个特征图将S2 中所有特征图为输入，C3 层有共(5×5× 3+1)×6+(5×5×4+1)×6+(5×5×4+1)×3+(5×5×6+1)×1=1516个可 训练参数，10×10×1516=151600个连接。

⑤S4 层是一个子采样层，由16个5×5大小的特征图构成。特征图中的每个单元与 C3中相应特征图的2×2邻域相连接，跟C1 和S2 之间的连接一样。2×2的小方框，每个 小方框有1个参数加上1个偏置， S4 层有(1+1)×16=32个可训练参数(每个Feature Map 有1个参数和一个偏置)。对于S4 层的每一个图的每一个点，连接数是2×2+1=5, 总共是(2×2+1)×5×5×16=2000个连接。

⑥C5层是一个卷积层，有120个特征图，Feature Map大小为1×1。每个单元与 S4 层的全部16个单元的5×5邻域相连。由于S4 层特征图的大小也为5×5(同滤波器一 样),故C5 特征图的大小为1×1,构成了S4 和 C5 之间的全连接。C5 与 S4 之间含有 (5×5×16+1)×120=48120个参数，(5×5×16+1)×120=48120个连接。

⑦F6 层有84个单元，与C5 层全相连，有120×84+84(偏置)=10164个可训练参 数，10164个连接。如同经典神经网络，F6 层计算输入向量和权重向量之间的点积，再 加上一个偏置，然后将其传递给Sigmoid 函数产生单元i 的一个状态。

⑧最后，输出层由欧式径向基函数 (Euclidean Radial Basis Function) 单元组成，每 **类一个单元，每个有84个输人。**



(7)卷积神经网络背后的思想。

卷积神经网络一般用于图像识别。对于100×100像素构成的图片，输入数据是100×

100个像素构成的向量(如果是彩色图像，那么每个像素包含三个原色，输入向量的大小 是100×100×3)。如果隐藏层的神经元数量和输入层一样多，需要计算(100×100)× (100×100)=108个权重。即便利用并行计算或者分布式计算，都是繁重的任务，有必要 对问题的解决进行巧妙的规划，加快计算过程。

①局部感知。首先，当我们观察图片的某个局部区域时，距离较远的像素对视觉的影 响是很小的，可以忽略不计。局部感知指的是隐藏层的神经元，只需和输入层的和它位置 近邻的像素发生关联即可，比如一个隐藏层的神经元与输入层的10×10区域的像素发生 关联。对于这样的隐藏层和输人层的连接模式，神经元之间连接的数量降低了，只需计算 (100×100)×(10×10)=10⁶个权重即可。

②权重共享。当图片上的像素、边缘、物体发生平移或者位置变化，我们还是可以 理解这个图片的，也就是我们的视觉不依赖这些对象在图片上的绝对位置。这表示我们 训练出来的权重(比如10×10的卷积核)可以用于图片的各个位置。换句话说，在一 个10×10的邻域内学习到的特征可以作为一个筛选器，应用到整个图片范围。 一个卷 积核可以理解为一个特征，卷积神经网络可以通过设计多个卷积核提取更多特征。

③池化。当网络结构很复杂，但是样本数量不够大时，深度神经网络容易发生过拟合 的问题 (Over Fitting)。池化 (Pooling) 或者子采样 (Subsampling), 通过 n×n 的小区 **域的汇总，磨平该区域的突出特征，避免过度学习问题**。

常见的图像识别技术，就是通过多阶段的卷积层、池化层的组合，构造深度卷积神经 网络实现的。各个阶段输出越来越复杂、抽象级别越来越高的图片特征，最后作为输入变 量，输入到一般的神经网络层，进行分类训练。

卷积神经网络针对图像识别问题，构建了面向二维矩阵的神经网络技术。无须人工寻 找图像特征，而是由神经网络自己从数据中找出特征，卷积层越多.能够识别的特征越抽 象。如果我们要训练神经网络从图片中识别猫，只需把大量的猫的图片交给神经网络，它 就能够找出猫的特征。

(8)循环神经网络 (Recurrent Neural Network)。

对于自然语言处理、语音/视频识别等应用，样本出现的时间先后顺序非常重要，使 用 CNN 神经网络进行分析不适合。在这样的应用场合，循环神经网络 (RNN) 是更好的 方案。人们使用循环神经网络来对时间关系进行建模。

和普通的前向反馈 (Feed Forward) 神经网络相比， RNN 只是在中间隐藏层多了一 个循环的圈而已(如图5-30 (a) 所示),这个圈表示上次隐藏层的输出作为这一次隐藏 层的输入，也就是一个神经元在时间戳t 的输出，下一时刻戳t+1 作为输入作用于自身， 图 5 - 3 0 (b) 展示了有两个隐藏层节点的RNN。

直观来讲，这样做的目的是希望让网络下一时刻的状态与当前时刻相关，即我们需要 创建一个有记忆的神经网络。因此，RNN 是适合处理时序数据的神经网络模型，用于训 练时间序列数据模型，然后用于分类和预测的任务。

为了方便分析，我们常常将 RNN 在时间上进行展开.得到如图5-31所示的结构。 (t+1) 时刻的网络的输出结果O(t+1) 是该时刻输入和“所有历史”共同作用的结果。正

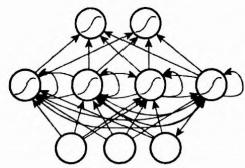


Output

Hidden layer

Input



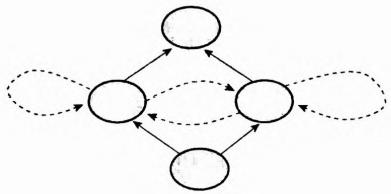


Output Layer

Hidden Layer

**Input Layer**

(a)

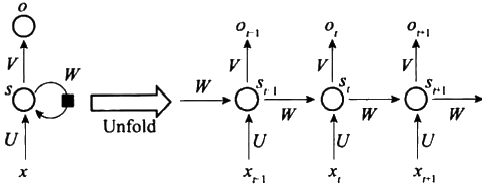


(b)



**图** **5** **-** **3** **0** **RNN** **的结构(虚线箭头表示** **Edge** **to** **Next** **Time** **Step)**

是这种建模方式，使得RNN 很适合对时间序列进行建模。图5-31 (b) 是通过两个时间 步来展开图5-30 (b) 的网络表示，将连接以无环图的形式可视化。我们可以通过图5- 31(a) 和 (b), 加深对循环神经网络的理解。

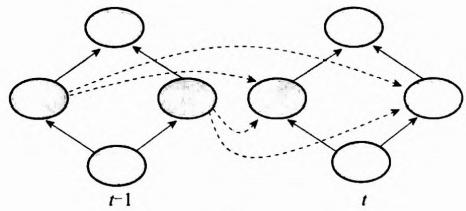


Output

Hidden layer

Input

(a)



(b)

**图** **5** **-** **3** **1** **RNN 在时间上进行展开(虚线箭头表示** **Edge to Next Time Step)**

隐藏层神经元s, 的输入包含两个部分，一个是x₁, 一个是s-1 。t 时刻RNN 的 o, 的计 算过程为 ：

①t 时刻.隐藏层神经元的激活值为s,=f(u×x,+w×s-1+b₁) 。 注意这里的s-1 表

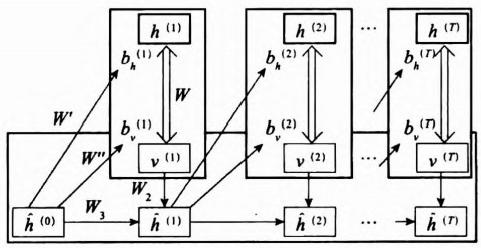
示t-1 时间步隐藏层的输出。

②t 时刻，输出层的激活值为o,=f(v×s,+b₂)。

循环神经网络是一种深度神经网络。它的深度不仅表现在输入和输出之间，而且表现 在不同的时间步之间，每个时间步可以被认为是一个层。

循环神经网络使用跨时间步的反向传播算法，即沿着时间的反向传播 (Backpropaga- tion Through Time) 算法，进行端到端的训练。

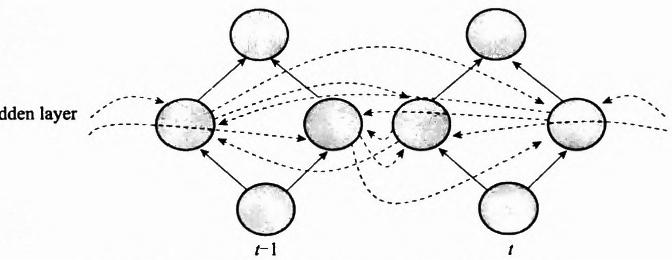
RNN 的应用很广泛，其中 ICML 会 议 (International Conference on Machine Learn- *ing) 上的一篇论文* *Modeling Temporal Dependencies in High-Dimensional Sequences:* Application to Polyphonic Music Generation and Transcription 提出，结合RNN 和 RBM 的神经网络模型，用来对复调音乐进行建模，建好的模型用于产生复调音乐。图5-32展 示了这个RNN-RBM 模型，其中每个粗线框表示一个RBM, 细线框则表示按照时间展开 的 RNN。



**图5-32** **RNNRBM 网络结构**

(9)双向RNN(Bidirectional Recurrent Neural Network,BRNN)。

RNN 通过参考历史信息对样本的时间关系进行建模。 一个自然的问题是，RNN 是否 可以参考未来信息呢?答案是肯定的。这就是双向RNN。 图5-33沿着时间步，展示了 最简单的一种双向RNN 网络结构。可以看出，其隐藏层同时使用历史和未来的信息进行 预测。

Output

**Input**

**图5-33** **一种沿时间步展开的通用的双向RNN 结构**

双向 RNN 使用沿着时间的反向传播算法进行端到端的训练。只是需要在训练样本的 开始和结束部分(训练样本具有时间关系)给予特殊处理。在t=1 步，前向的隐藏层状 态输入 (Forward State Input) 是未知的， t=T 步，后向的隐藏层状态输入 (Backward

State Input) 也是未知的，都可以设定为某个值。

(10)LSTM(Long Short Term Memory) 网络。

LSTM 本质上是一种RNN 。RNN 在训练时，往往会遇到严重的梯度消失问题，也就 是误差梯度随着事件的时间差的大小快速下降。这是发生在时间轴上的梯度消失。理想情 况下，我们希望“所有历史”都对当前隐藏层的节点的预测产生作用。实际上，这种影响 只是维持了若干时间步。换一种说法，就是后面时间步的错误信号，往往不能回到足够远 的过去，像更早的时间步一样影响网络，这使得神经网络难以学习远距离(时间上)的影 响 。Yoshua Bengio 在其高引用论文 Learning Long-Term Dependencies with Gradient Descent is Difficult 中，对梯度消失问题进行了深入论述。

1997年Sepp Hochreiter 和 Jurgen Schmidhuber 提出的长短期记忆 (Long Short Term Memory,LSTM) 模型，很好地解决了这种梯度消失问题。 LSTM 通 过 门 (Gate) 开关，实现时间上的记忆功能，并防止梯度消失。使用LSTM 模块后，深度 神经网络的误差，从输出层反向传播回来时，可以用模块的记忆单元记录下来。于 是 ，LSTM 可以记住一段时间内的信息，能够学习到前后事件之间更长时间范围的依 赖关系。

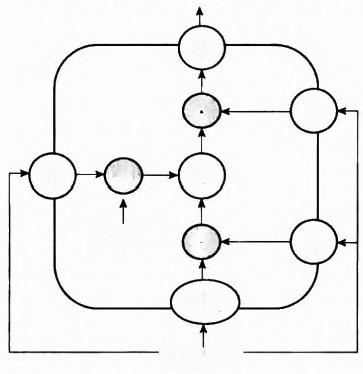
LSTM 适合对时间序列数据进行预测，特别是在数据中存在较长时间范围内的重要的 事件之间依赖关系时。在这样的场合下，LSTM 比其他序列学习 (Sequence Learning)方 法，比如RNN 、隐马尔可夫模型 (Hidden Markov Models,HMM) 等都具有更好的 性能。

2009年基于LSTM 的模型，获得ICDAR 手写体识别比赛的冠军。2013年Graves 等 使用双向LSTM, 在音素识别测试数据集 (TIMIT Phoneme Recognition Benchmark) 上 ， 获得17.7%错误率 (Error), 是当时的最好纪录。

LSTM 有很多变种，下面介绍最简单的一种。一个Cell 由三个门Gate(Input,For-

get,Output) 以及一个Cell 单元 (Cell Unit) 组成。Gate 使用Sigmoid 激活函数，Input 和Cell 状态通常使用Tanh 函数来进行转换。

对照图5-34,一个LSTM Cell可以定义成如下一系列等式。



h,

0,

*f,* c,

(c:₋1)

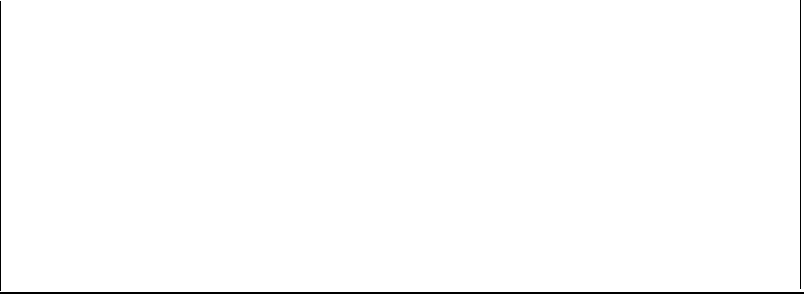
c\_in,

(x,h,-1)

i,

**图5-34** **LSTM** **的一个单元** **(Cell)**

文



门 (Gate) 的转换

i,=g(Wnx,+W₀h,-1+b;)

f,=g(Wx,+Wh₁h,-1+b)

0,=g(Wmx,+Wh,-1+b₀)

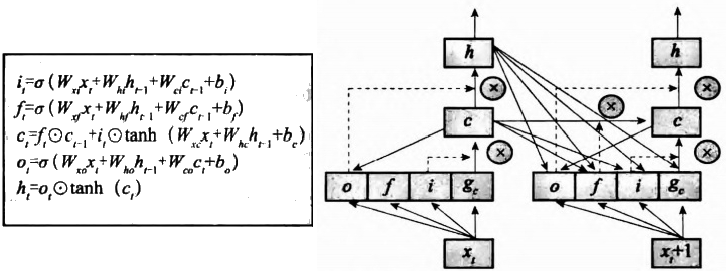
输人的转换 (Input Transform)

c\_in,=tanh(Wπx,+Wh;-1+bei)

状态更新 (State Update)

c,=f ·C₁-1+i, ·c\_in, (注意，c可以看作LSTM 的存储器) h,=0,·tanh(c₁)

可见，由于有了门控机制 (Gating Mechanism),Cell 在工作时可以保持较长时间段 的信息，并且在训练时保护Cell内部的梯度不受不利变化的影响，减少梯度消失效果。图 5-35展示了一个稍微复杂一点的LSTM Cell模型，这个模型和图5-34展示的模型的区 别是Input 门 、Forget 门的激活函数参考了上一个时间步的Cell 的状态C₁-1,Output 门 的激活函数参考了本时间步Cell 的状态C₁。



**图5-35** **另一个** **LSTM Cell 模型**

一个 LSTM 网络是包含LSTM 基本单元(当然也可以包含常规的人工神经元)的人 工神经网络。LSTM 网络可以使用沿着时间的反向传播算法 (Back Propagation Through Time) 进行训练，从输出层反向传播回来的误差被 LSTM 基本单元的存储器捕获 (Trapped), 于 是LSTM 基本单元能够记住很长时间范围内的一些数据变化情况。解决长 程依赖问题，是 LSTM 优于RNN 的地方。

(11)双向LSTM 结构。

双向LSTM 网络的目标和双向RNN 是一致的，那就是提供一种参考当前信息的较长 时间段内的历史和未来信息(当前事件的更长时间段内的上下文环境)的机制。其网络结 构和双向RNN 类似，区别在于其基本构造单元换成了LSTM Cell。

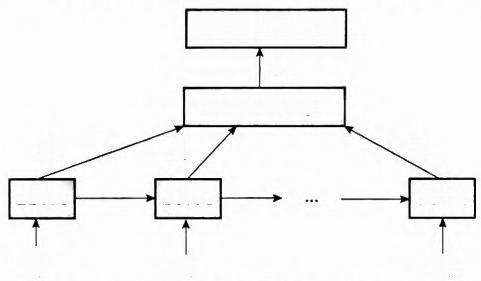
(12)RNN 和 LSTM 的应用。

RNN 和 LSTM 以其强大的时间关系建模能力应用到很多场合，包括时间序列的预测 (Time Series Prediction)、计算机音乐创作 (Computer Composed Music)、节奏学习 (Rhythm Learning)、语法学习 (Grammar Learning)、机器翻译 (Machine Translation)、 基于字符的 LSTM 语言模型 (Character-Level Language Model)、语音识别 (Speech Recognition)、自动给图像加描述 (Image Captioning,也称为图文转换)、手写体识别 (Handwriting Recognition)、机器人控制 (Robot Control)、人体动作识别 (Human Ac-



tion Recognition)、蛋白质同源性检测 (Protein Homology Detection) 等。

比如，deeplearning.net 网站提供了一个LSTM 的应用教程，通过创建一个LSTM 神 经网络模型，实现对电影评论数据集 (Movie Review Dataset) 的情感分析。这个模型由 一个LSTM 层、一个平均池化层以及一个 Logistic 回归层构成，如图5-36所示。



**Logistic Regression**

h

Mean Pooling

h

LSTM

*X₁*

h

LSTM

h。

LSTM

X

*x。*

**图5-36** **用于情感分析的基于LSTM的神经网络**

**8.** **深度神经网络模型训练的加速**

深度神经网络模型参数多，计算量大。当训练数据的规模很大时，需要消耗很多计算 资源，耗费相当长的时间，才能对模型进行训练。如何提高深度神经网络模型训练的效 率，是一个紧迫的问题。主要的技术手段简单介绍如下。

( 1 ) 利 用GPU 加快深度神经网络的训练。深度神经网络的训练过程可以表达成矢量 运算的形式。GPU(Graph Process Unit, 图形处理器)拥有上千个处理核心，可以将矢 量运算并行化执行，大幅缩短计算时间。利用GPU 来训练深度神经网络，可以充分发挥 其众核处理器的计算能力，提高神经网络模型的训练速度，在使用大数据进行模型训练时 尤显其性能优势。

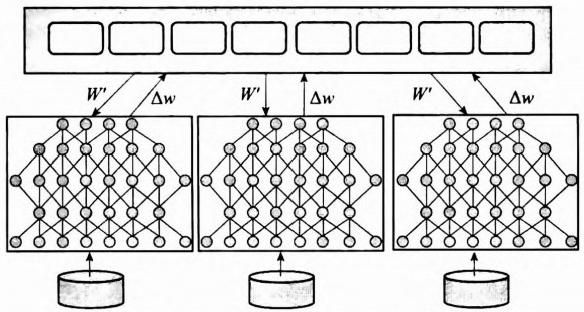
(2)数据并行。数据并行是对训练数据进行切分，采用多个模型副本，在各个分片数 据上并行训练。为了最后得到一个统一的模型，需要借助一个参数服务器，完成主模型和 各个模型副本的参数交换。在训练过程中，各个训练过程相互独立，训练的结果表现为模 型的参数变化量△W, 由各个训练过程汇报给参数服务器，由参数服务器对模型进行更新 W′=W-η·△W, 并且把更新以后的模型参数W′ 分发给各个训练过程，以便它们从新的 起点开始继续进行训练，如图5-37所示。

*W′=w-η△w*

Parameter Server

Model Replicas

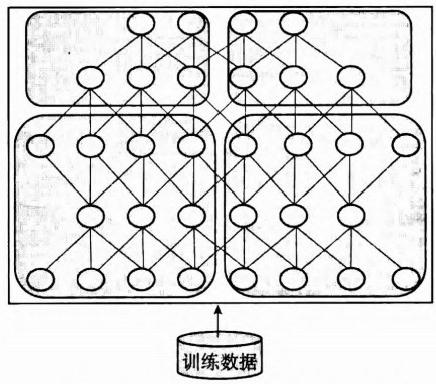
Data Shards



**图5-37** **深度神经网络模型的训练-数据并行**

(3)模型并行。模型并行将模型拆分成几个分片，由几个训练单元分别负责，共同协

作，完成训练。当一个神经元的输入包括来自另一个训练单元上的神经元的输出时，会产 生通信开销，如图5-38所示。



**图5-38** **深度神经网络模型的训练-模型并行**

当数据并行的训练程序太多时，需要减小学习率，以保证训练过程的平稳。当模型并 行的模型分片太多时，神经元输出值的交换量会急剧增加，训练效率将大幅下降。数据并 行和模型并行都不能无限地扩展，结合使用模型并行和数据并行，是一种可行的方案。

(4)CPU 集群。利用CPU 集群，进行深度神经网络模型训练，是工业界常用的解决 方案。它基于模型可分布式存储、参数可异步通信的特点，利用大规模分布式计算集群的 强大计算能力，加快深层模型的训练。CPU 集群的基本架构，包括用于执行训练任务的 工作节点 (Worker) 、 用于分布式存储分发模型的参数服务器 (Parameter Server), 以 及 用于协调整体任务的主控程序 (Master)。

CPU 集群方案适合于训练GPU 内存难以容纳的大模型以及稀疏连接神经网络。目前 结合GPU 计算和集群计算技术构建GPU 集群，正成为加快大规模深度神经网络训练的新 的解决方案。

(5)专用的机器学习处理器。为了进一步提高机器学习的训练性能， 一些公司和研究 机构开发了专用的处理器，包括中科院计算所的寒武纪处理器及 Google 公司的TPU (Tensor Processing Unit) 等，这些处理器在性能上优于通用CPU 或 者GPU, 功耗则低 得多。

**5.2.12** **特征选择**

在机器学习中，特征选择起到至关重要的作用。特征选择得好，可以对数据量进行有 效精简，提高训练速度，这点对于复杂模型来讲尤其重要。此外，特征选择可以减少噪声 特征 (Noise Feature), 提高模型在测试数据集上的准确性，防止过拟合 (Over Fit) 以及 欠拟合 (Under Fit) 情况的发生。一些噪声特征会导致模型出现错误的泛化 (Generaliza- tion),从而使得模型在测试集上表现较差。另外，从模型复杂度看，特征越多，模型越复 杂，越容易发生过拟合的情况。常用的特征选择方法有互信息、信息增益、卡方检验等

方法。

1. 信息增益法

信息增益法根据某特征项 (Feature) 为整个分类能够提供多少信息量，衡量该特征 项的重要程度，从而决定对特征项的取舍。某个特征项的信息增益指的是有该特征或没有 该特征时，为整个分类所能提供的信息量的差别，其中，信息量的多少由熵来衡量。

熵可以视为描述一个随机变量的不确定性的量。不确定性越大，熵越大。假设有一个 随机变量，它的可能取值有 n 个，分别是x₁,x₂, … ,xn, 取得每个取值的概率分别是P₁, P₂,…,P, 那 么X 的熵定义为：



对于分类系统来讲，类别C 是一个变量，它的可能取值是C,C₂,…,C,, 每个类别出现 的概率为P(C₁),P(C₂),…,P(Cₙ),n 是类别的总数。那么分类系统的熵可以表示为：



一个特征X, 特 征X 被固定时的条件熵为H(C|X), 如 果X 的取值为x₁,x2, … ,xn, 那 么

H(CI X)=P₁H(C|X=x₁)+P₂H(C|X=x₂)+…+P,H(C|X=xn)



以文本分类为例，我们现在有一个特征T ( 一 个词项Term), 要计算它的信息增益来 决定它是否对分类有大的帮助，以便保留该特征项。需要考察不考虑任何特征时文档的 熵，也就是没有任何特征时做分类，我们有多少信息。再考察考虑了该特征后，我们能有 多少信息。两者之差就是该特征给我们带来的信息。

一 般T 的取值为T 出 现 和 T 不出现两种，分别记为t 和E, 则 有

H(C|T)=P(t)H(C|t)+P(z)H(C|7) 信息增益：





公式的解释如下：P(C;) 为类别C, 出现的概率，用1除以类别总数来计算； P(t) 为 特 征 T 出现的概率，用出现过T 的文档数除以总文档数来计算；P(C;|t) 为出现T 时类别C; 出现的概率，用出现了T 并且属于类别C; 的文档数除以出现了T 的文档数来计算。

信息增益法考察特征对整个系统的贡献而不能具体到某个类别上，于是它只适合用来 做“全局”的特征选择，所有的类使用相同的特征集合。信息增益法无法做“本地”特征 选择，也就是每个类别有自己的特征集合的场合，比如有些词对某些类别很有区分度，对 另一个类别则无关紧要。

2. 卡方检验法

卡方检验是统计学的基本方法。它的基本思想是通过观察实际值与理论值(预期)的 偏差来决定理论正确与否。

首先提出原假设，假设两个变量确实是独立的，比如在文本分类中，我们关心一个特



征(也就是 一个词Term, 简 称t) 与一个类别C 是否相互独立。如果相互独立，那么说明 词 Term 对类别C 没有区分作用，也就是我们无法根据t 是否出现来判断一篇文档是否属 于类别C。

提出原假设后，根据观察实际值与理论值的差别(原假设成立情况下其应该有的值), 如果这个差别足够小，我们可以认为误差是由于测量手段不够精确导致或者偶然发生的， 两者确确实实是独立的，此时就接受原假设。如果这个差别足够大，则这个误差不太可能 是偶然产生或者测量不精确所致，词t 和类别C 两者实际上是相关的，也就是我们否定了 原假设，接受了备择假设。

在这里，换句话说，可以认为卡方检验中使用特征与类别间的关联性来进行这个量 化，关联性越强，特征得分越高，该特征越应该保留。

卡方统计量x² 的计算公式为  。一旦提供了多个样本的观察值x₁,x₂,…, x₁,…xn, 代人该公式，即可求卡方值。用这个值与事先确定的阈值比较，如果大于阈值 (偏差很大),就认为原假设不成立。

根据文档集 (N 个文档)的情况，创建列联表(见表5- 18)。行方向为词t 是否出 现，列方向为文档是否属于类别C。

**表5-18** **列联表**

|  |  |  |  |
| --- | --- | --- | --- |
| 特征选择 | 1.属于类别C | 2.不属于类别C | 总计 |
| 1.包含特征t | A | B | A+B |
| 2.不包含特征t | C | D | C+D |
| 总计 | A+C | B+D | N |

包含特征t, 并且属于类别C 的文档数量的期望值为： E₁=(A+C)×(A+B)/N

D₁₁=(A-En)²/E₁1 卡方统计量：



计算出卡方统计量后，和阈值做比较，决定是否拒绝原假设。其他特征也用同样的方 法处理，按照卡方统计量进行由大到小的排序，然后对特征进行选择。

表5-19为卡方分布在自由度为1时的显著性阈值。

**表5-19**

|  |  |
| --- | --- |
| p | x²critical value |
| 0.1 | 2.71 |
| 0.05 | 3.84 |
| 0.01 | 6.63 |
| 0.005 | 7.88 |
| 0.001 | 10.83 |

假设我们计算出的x² 值是283,它远远大于10.83,我们可以在0.001的显著性水平 下拒绝原假设，认为t和C 并不独立，而是具有很强的相关性。

3. 互信息法

互信息用来评价一个事件的出现对于另一个事件的出现所贡献的信息量，它的基本定

义公式为：



在文本分类中，我们关心一个特征(也就是一个词Term, 简称t) 与一个类别C 。把 这个公式运用到文本分类中。我们有：



式中，U,C 都是二值随机变量，当文档包含词项t 时 ，U 的取值为e,=1, 否则e=0; 当 文档属于类别C 时 ，C 的取值e.=1, 否则e.=0。 用最大似然估计时，上面的概率值都是 通过统计文档中词项和类别的数目来计算的。





式中，N 表示全部数据中两个事件同时出现的概率；N 表示全部事件出现的次数；No. 则

表示N₀+N (见表5-20)。

**表5-20**

|  |  |  |  |
| --- | --- | --- | --- |
| 特征选择 | 1.属于类别C | 0.不属于类别C | 总计 |
| 1.包含特征t | N1 | N | N. |
| 0.不包含特征t | Nol | Noo | No. |
| 总计 | N. | N.。 | N |

针对“某个特征是否出现”和“是否属于某个类别”两个事件进行计算，然后保留得 分较高的特征。

5.3 主 流 数 据 深 度 分 析 工 具



本节将介绍主流的机器学习和数据挖掘软件包，包括 Hadoop 平台上的 Mahout 软件 包，Spark 平台上的MLlib 软件包，开源软件系统Weka,R, 商用软件系统SPSS,Mat-

lab, 以及深度学习软件包TensorFlow,Caffe 等。

5.3.1 Mahout系 统

Mahout 是 Hadoop 大数据平台上的开源机器学习软件包。Mahout 提供了在大规模集 群上对大数据进行深度分析的能力。主流的数据挖掘和机器学习算法不断在Mahout 平台 上实现，包括聚类、分类、协同过滤 (Collaborative Filtering, 用于推荐)以及频繁集挖 掘等众多的算法。

MapReduce 计算模型固有的局限性(在Stage之间和Job 之间，中间结果经过磁盘进行传 递),这些算法的执行效率不高。2014年Mahout 社区决定把Mahout 代码迁移到其他更 高性能的大数据平台上运行，包括Spark 和 H2O 。Mahout 不再接受使用MapReduce 编程 模型实现的代码，原先已经用MapReduce 模型实现的代码仍然保留，并且由Mahout 社区 提供维护和支持。

迁移到Spark 平台的原因是Spark 提供了更加灵活的编程模型和更加高性能的执行引 擎。针对 Spark 平台开发的机器学习算法，可以充分利用Spark 的内存数据处理能力，提 高机器学习算法的执行效率。Mahout 经过改写以后，成为一个更加通用的软件包，并不 依赖于Spark, 也可以运行在其他底层大数据处理系统上，比如 H2O 。H2O 是由初创公 司Oxdata 开发的内存数据处理引擎，用于从HDFS 上加载数据集，执行各种机器学习和 统计分析算法。Mahout 和 Spark平台上的MLlib 都是机器学习软件包，某种程度上提供 了重复的功能，但是有选择对用户来讲终究是一件好事。

**5.3.2 Spark MLlib 系** **统**

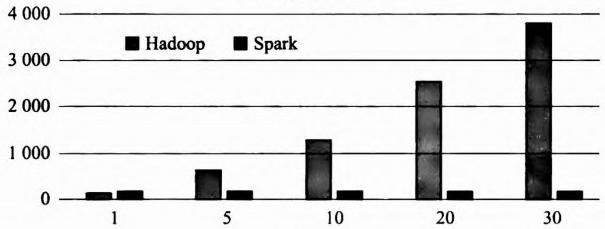
MLlib 是 Spark 大数据平台上的机器学习程序库。这个库包含很多通用的机器学习与 数据挖据算法，包括分类、回归、聚类、协同过滤、降维 (Dimensionality Reduction), 还包含底层的优化原语 (Optimization Primitive) 以及数据分析流水线应用程序编程接口 (API), 在 Spark 平台上对数据进行深度分析的编程变得更加容易。

MLlib 包含两套API, 分别是spark.mllib 和 spark.ml, 前者包含老版本Spark 原有 的 RDD①上的API, 后者是基于DataFrame 数据抽象层的高层的API。

由于运行在Spark 平台上，利用Spark 的高性能和大规模数据处理能力，某些数据分 析任务比Hadoop 上的MapReduce 实现获得10倍(数据保存在磁盘中)到100倍(数据 完全存放在内存中)的性能提高。

MLlib 的实现者在2010年对Spark 平台上的Logistic 回归分析程序以及 Hadoop 平台 上的Logistic 回归分析程序进行了性能比较。他们在 Amazon 云平台上使用了20个 “ml.xlarge”EC2 虚拟机实例，每个实例有4个核 (Core), 实验数据集是29GB。结果如 图5-39所示，从中可以看出Hadoop 平台和 Spark 平台上机器学习算法性能的差异。

**Running Time(s)**



Number of Iterations

**图5-39** **Hadoop平台和Spark平台上的Logistic 回归分析性能比较**



①请参考第13章 “Spark 及其生态系统”。

数据挖掘和机器学习程序可以用Java,Python 或者Scala 语言编写。下面是使用Sca- la语言编写的K-Means 聚类程序实例。①

|  |
| --- |
| //K-Means Scala程序 //导人必要的库  import org.apache.spark.mllib.clustering.KMeans  import org.apache.spark.mllib.linalg.Vectors  //装载数据、解析数据  val data=sc.textFile("data/mllib/kmeans\_data.txt")  val parsedData=data.map(s =>Vectors.dense(s.split(”).map(\_.toDou- ble)))  //使用K-Means 聚类算法把数据聚类成两个类簇 val numClusters =2  val numIterations =20  val clusters =KMeans.train(parsedData,numClusters,numIterations) //计算 WithinSetSumofSquaredErrors,评估聚类效果  val wSSSE=clusters.computeCost(parsedData)  println("Within Set Sum of Squared Errors="+WSSSE) |

**5.3.3 Weka 系** **统**

Weka的全名是怀卡托智能分析环境 (Waikato Environment for Knowledge Analysis)。 Weka是一款免费的基于Java 语言的开源机器学习和数据挖掘软件。Weka的主要开发者来自 新西兰的University of Waikato, 该软件的缩写Weka正好是新西兰的一种独有的鸟的名字。

在 Weka 平台上已经实现了数据预处理、分类、聚类、回归、关联规则分析等机器学 习和数据挖掘算法，用户不仅可以使用 Weka 的工作台可视化地对数据进行分析，而且可 以调用Weka 的 Java 开发包，编写自己的数据分析软件。

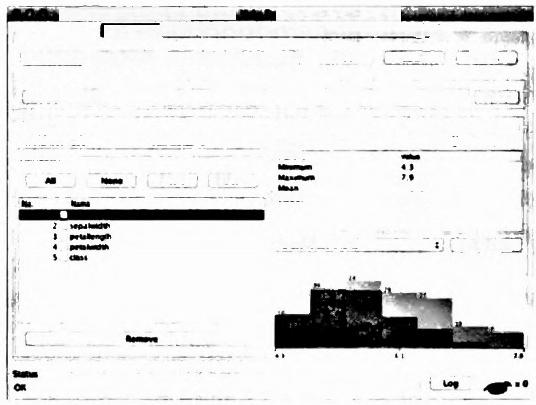
Weka Explorer (探索者界面)是 Weka 最简单的使用界面(见图5-40)。所有的 Weka功能都能在这个界面中通过点击鼠标和表单填写来使用。由于很多选项都预设了常 用的默认值，使用户可以最小的代价取得结果。在这个界面上，有六个标签，分别是数据 预处理 (Preprocess) 、 分 类 (Classify) 、 聚类 (Cluster) 、 关联分析 (Associate) 、 属性选 择 (Select Attributes) 和可视化 (Visualize)。

Weka Knowledge Flow (知识工作流)的主体是一个设计画布。用户从工具栏中选择 Weka组件并将其置于设计画布上，连接成一个数据处理和分析的工作流。用户通过可视 化的编程，可以实现灵活和复杂的数据分析功能。

知识工作流界面共有8个标签，分别是Data Sources 用于选择数据源，Data Sinks用 于保存结果，Filters 用于选择过滤器， Classifiers 用于选择分类器， Clusterers 用于选择

①<https://spark.apache.org/docs/1.1.0/mllib-clustering.html#examples.>



Cue Aou a

Om 奥 OoD8. e Lndn .

C

Pher

Cm

Ak

sshected nnte

Crw nan

Mtanon i

tans:150

Ma seplone

Ie n

g Dueet 1s

Ar :5

n hun

Am

41

De 0.4

Cless cass Nem

WdN

**图5-40** **Weka** **Explorer** **界面**

聚类器，Associations 用于选择关联规则算法，Evaluation 用于指定评估方法，以及Visu- alization 用于选择将结果进行可视化的组件。

在实际研究工作中，研究者可能需要处理好几个数据集，计算量非常大，这时候就需 要实验者界面 (Weka Experimenter) 来解决问题了。实验者界面允许使用多种算法对多 个数据集进行操作，并且支持分布式计算。用户可以通过 Setup 标签页，选择实验结果的 输出方式。通过Run 标签页，开始或者结束挖掘和分析。通过Analyze 标签页，查看分析 结果。在Setup 标签页的Iteration Control 面板上，用户可以设定验证次数，以及使用多 个算法遍历多个数据集的顺序。

和 Weka 类似的数据挖掘软件包是RapidMiner(<https://rapidminer.com/>), 其前身 是 YALE 。RapidMiner 是一款流行的开源软件包，它不仅提供了一个GUI (图形用户界 面)的数据处理和分析环境，而且提供了Java API, 以便用户将其分析功能嵌入到Java 应用程序中。

**5.3.4 R 系统与语言**

R 是一套开源免费的软件环境，用于对数据进行统计分析 (Statistical Computing) 和 可视化 (Graphical Display) 。R可以运行在Unix,Windows 以及MacOS 操作系统上。

R 的特性包括：(1)有效的数据存储和管理；(2)对数组、矩阵等数据对象进行计算 的一套操作原语；(3)统计分析工具集；(4)对数据分析结果的可视化能力；(5)简便而 有效的编程语言(称为S), 支持数据的输入输出，可实现分支、循环、用户自定义函数 等功能。CRAN 为 Comprehensive R Archive Network的简称，它除了收藏了R 的可执行 版本、源代码和说明文件外，还收录了开源社区贡献的各种扩展软件包。新的扩展软件包 使得 R 可以支持分布式计算 (Distributed Computing), 处理更大的数据集。目前，全球 有超过100个CRAN 镜像站点。

为了使用R 环境，用户需要了解 R 的数据结构和编程语言。 R 的基本数据类型包括 整数 (Integer) 、 小 数 (Decimal) 、 复 数 (Complex) 、 逻 辑 真 假 (Logical) 、 字 符 串



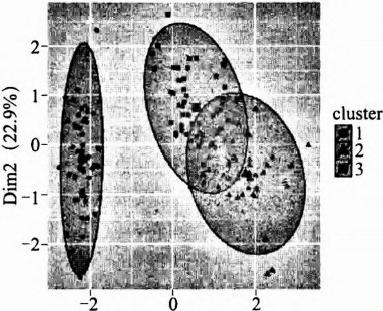
(Character) 。在此基础上，可以构造向量 (Vector) 、 矩阵 (Matrix) 、 列表 (List) 以及 数据框架等数据对象。其中，向量是同类型的数据成员 (Member) 构成的一个有序序 列，矩阵把数据成员组织成一个二维表的形式，列表则是一个通用的向量，可以包含其 他类型的对象，数据框架用于存储表格，表格可以认为是一个等长的向量(行向量)的

列表。

在上述数据对象之上，用户可以使用现成的分析软件包，对数据进行统计分析。主流 的统计分析方法，以及机器学习和数据挖掘方法，都已经有开源的实现。只有在现成的软 件包不能满足分析要求的情况下，才需要用户编写自定义函数。

R 软件包具有强大的分析结果可视化功能。图5-41显示了对数据进行K-Means 聚 类 分析以后的数据可视化结果。

Cluster plot



(22.9%(

Dim1(73%)

**图5-41** **R的分析结果可视化实例**

资料来源： <http://www.sthda.com/english/wiki/determining-the-optimal-number-of-clusters-3-must-known-methods>- unsupervised-machine-learning.

R 软件包自2014年以来逐渐成为数据分析者的首选工具。有分析认为，全球有超过 百万的分析师(包括最终用户)寻找容易使用的大数据分析工具，R 软件包有可能替代商 用的SAS 及 SPSS 分析软件，成为人们的最佳选择。

**5.3.5 SPSS与Matlab**

**1.SPSS**

SPSS 是著名的统计软件包。1968年3名斯坦福大学的学生研发了最初的 SPSS 系 统 ， 并且于1975年成立了SPSS 公司，目前SPSS 被 IBM 公 司 收 购 。SPSS 可以运行在 Win- dows,MacOS,Unix 等操作系统上，我们常用的 SPSS 的 版 本 是Windows 上 的 SPSS。

SPSS 的功能包括数据管理、统计分析、图表生成、输出管理等。 SPSS 是 Statistical Pack- age for Social Science的缩写，由于SPSS 不仅适用于社会科学，而且有了更加广泛的应用 范围，其名字已经改为Statistical Product and Service Solutions, 缩写仍然是SPSS。

SPSS 是一个软件家族，这个家族中包括 SPSS Statistics,SPSS Modeler,SPSS Col- lection 等软件，分别实现统计分析、预测性分析、数据收集等工作。

SPSS Statistics是专业的统计分析软件，它具有方便易用的图形用户界面 (Graphic User Interface,GUI), 得到广泛应用，受到用户喜爱。SPSS Statistics以两种方式运行， 分别是窗口方式和命令行方式。在窗口方式下，用户通过窗口、菜单、对话框等界面对数 据进行分析。在命令行方式下，用户通过命令和参数调用各种数据分析例程。用户还可以 编写自己的数据分析程序。SPSS 公司把 SPSS Statistics的分析功能组织成不同的模块， 用户可以根据需求从中选择一个或多个模块，来实现所希望的分析功能。

SPSS Statistics软件拥有自己的分析语言Syntax 。SPSS 支持的数据类型包括数值型、 日期型、字符串型等。SPSS 的分析语言支持数学运算、关系运算、逻辑运算。数据的组 织方式类似于 Excel 的二维表。SPSS已经内置支持各种数据操作和分析功能的函数库， 包括算术函数、统计函数、随机变量函数、逆分布函数、累计分布函数、逻辑函数、日期 时间函数、缺失值函数、字符串函数、转换函数等。要分析的数据和分析结果都保存在数 据文件中，数据文件由数据结构和具体数据组成。数据结构是对数据的模式的定义.比如 包含什么变量，每个变量是什么类型，每个变量使用什么标签等。

SPSS Modeler 是一个预测性的数据分析平台，为个人和企业提供预测性的分析结果。 它不仅可以分析结构化数据(比如用户、产品、地理信息等),而且可以分析非结构化数 据(比如文本、电子邮件、社交媒体数据等)。 SPSS Modeler 使用统计方法、文本分析方 法、实体分析和社交网络分析方法等，揭示数据中隐藏的模式，使得用户对数据的了解更 为深入，其中的实体分析和社交网络分析方法可以揭示个人和群体的社交行为模式。

SPSS Data Collection用于设计多样的调查问卷，以便获取抽样样本，快速收集有效的调 查数据，剔除不完整的数据反馈，生成各种统计报告。SPSS Data Collection 支持三种问卷调 查模式，包括基于Web的问卷调查、利用计算机辅助电话访谈的问卷调查以及离线的手持设 备辅助的问卷调查等。在商业活动中，企业希望尽快了解市场的现状与发展趋势，为营销决 策提供客观的数据支持。市场调查是运用科学的方法，有目的地系统收集有关市场的数据。 SPSS Data Collection正是支持这个功能的一整套软件方案，它帮助用户高效准确地获取他们 需要的信息，并对用户反馈的信息进行分析和总结。SPSS Data Collection 已经应用到消费 品 、IT/ 电信、汽车、金融/保险、医药/健康等众多领域。

**2.Matlab**

20世纪70年代后期，美国新墨西哥大学计算机系的系主任 Cleve Moler给学生讲授 线性代数课程，他让学生使用EISPACK 和 LINPACK 程序库。他发现学生用FORTRAN

编写接口程序很费时间，于是他利用业余时间为学生编写EISPACK 和 LINPACK 的接口 程序。Cleve Moler 给这个接口程序取名为Matlab, 是矩阵 (Matrix) 和实验室 (Labora- tory) 两个英文单词前三个字母的组合。在以后的数年里，Matlab 在多所大学里作为辅助 教学软件使用，并作为面向大众的免费软件，开始广为流传。

1983年，Cleve Moler 到斯坦福大学讲学，Matlab 深深地吸引了工程师John Lithe, 他敏锐地觉察到Matlab 在工程领域的广阔前景。同年，他和Cleve Moler,Sieve Bangert 一起，用C 语言开发了Matlab 第二代专业版，这一版同时具备了数值计算和数据图形化 的功能。1984年，Cleve Moler 和John Lithe 成立了MathWorks 公司 . 正式把Matlab 推 向市场，并继续进行Matlab 的研究和开发。

目前Matlab已经成为强大的支持科学计算、可视化以及交互式程序设计的数据分析 软件包。它将数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等 诸多强大功能，集成在一个易于使用的窗口环境中。Matlab 已经应用到数值分析、图像



处理、信号处理、金融建模、工程与科学绘图、控制系统的设计与仿真、通信系统的设计 与仿真等领域。

Matlab 的编程语言M (文件扩展名为M) 是一种高级的矩阵/数组操作语言，它包含 顺序/分支/循环控制语句、输人/输出函数等功能，是面向对象的编程语言。该语言基于 最为流行的C++ 语言创建，其语法特征与C++ 语言很相似，但是更为简单，非常便于 非计算机专业的科技人员使用，而且该语言可移植性好、可扩展性强，在科学研究及工程 计算等领域得到广泛应用。

除了内置基本函数之外， Matlab 还可以使用各种工具箱，实现丰富的分析和计算功 能。常用的工具包如表5-21所示。

**表5-21** **Matlab的常用工具箱**

|  |  |
| --- | --- |
| Matlab Main Toolbox//主工具箱 | Control System Toolbox//控制系统工具箱 |
| Communication Toolbox//通信工具箱 | Financial Toolbox//金融数据分析工具箱 |
| System Identification Toolbox//系统标识工具箱 | Fuzzy Logic Toolbox//模糊逻辑工具箱 |
| Higher-Order Spectral Analysis Toolbox//高阶谱 分析工具箱 | Image Processing Toolbox//图像处理工具箱 |
| Computer vision system toolbox//计算机视觉工具 箱 | LMI Control Toolbox//线性矩阵不等式工具箱 |
| Model predictive Control Toolbox//模型预测控制 工具箱 | μ-Analysis and Synthesis Toolbox//μ分析工具箱 |
| Neural Network Toolbox//神经网络工具箱 | (ptimization Toolbox//优化工具箱 |
| Partial Differential Toolbox//偏微分方程工具箱 | Robust Control Toolbox//鲁棒性控制工具箱 |
| Signal Processing Toolbox//信号处理工具箱 | Spline Toolbox//样条曲线工具箱 |
| Statistics Toolbox//统计分析工具箱 | Symbolic Math Toolbox//符号数学工具箱 |
| Simulink Toolbox//仿真工具箱 | Wavelet Toolbox//小波分析工具箱 |
| DSP system toolbox//DSP系统工具箱 |  |

Matlab 提供了其他语言的编程接口，支持用户使用其他语言编程，调用Matlab 的强

大数据分析和计算功能。目前， Matlab 提供了对 C/C+ 十 ，FORTRAN,JAVA 等语言

的支持。除了SPSS 和 Matlab 之外，市场上还有SAS 和 Statistica 等统计分析软件包供用 户选择。

**5.3.6 深度学习工具** **TensorFlow,Caffe**

**1.TensorFlow**

TensorFlow 是一款开源的深度学习软件包。 Google 公司开发了两代深度学习系统， 第一代称为 DistBelief, 第二代称为TensorFlow 。 这两套系统都已经应用在Google 的大量 产品中，包括语音识别、图像识别以及机器翻译等。 Google 开发第二代深度学习软件系 统的目的是增强软件的灵活性和执行性能。2015年，Google 宣布把 TensorFlow 机器学习 软件系统开源出来，供全世界的AI 工程师、学者以及编程人员使用，可以用于研究以及 实际产品的开发。

TensorFlow 具有灵活的架构，用户使用相同的API 开发程序，可以部署到不同的平 台上，包括移动设备、台式机以及服务器集群。TensorFlow 可以运行在CPU, 也可以运

行在GPU 上，利用GPU 的并行处理能力加快深度学习的过程。

TensorFlow 把计算过程表达成一个有向无环的数据流图 (Data Flow Graph)。图 中 的节点表示某种数学运算 (Mathematical Operation) 或者数据输入、数据输出、读写持 久化的变量等 (Persistent Variable)。图中的边表示节点之间的输人输出关系，也就是节 点间的多维数组(称为 Tensor, 或者张量)的通信传输。多维数组 (Tensor) 流过 (Flow) 一个流程图 (Graph), 这正是 TensorFlow 得名的来由。这种计算模型，类似于 Spark 或者 Dryad 的图计算模型。

用户定义机器学习的逻辑，软件架构则负责资源分配、任务调度以及容错等问题。数 据流图的调度采用异步并行处理的方式，即图的各个节点分配到计算资源上 (CPU), 然 后当进入该节点的边上有可用数据(即Tensor) 时，这些节点被异步执行.执行过程利 用了CPU 和 GPU 的并行处理能力。

目 前 ，TensorFlow 支持CNN,RNN 和 LSTM 等神经网络结构.这些网络结构在图 像识别、语音识别、自然语言处理、机器翻译等领域得到了广泛应用。

TensorFlow 以其强大的功能，被Google 自己和其他公司和广大开发者运用到不同的 应用中。比如 Google Brain研发了Smart Reply, 可以扫描 Gmail 邮件，并且自动生成回 复，供用户进行选择。TensorFlow 还运用到机器翻译、手写文字识别、为图片自动加标 题 (Auto Caption) 以及创作艺术作品等场合。

**2.Caffe**

Caffe是一个高效的深度学习软件框架，其作者是毕业于加州大学伯克利分校的贾扬 清博士。Caffe 使用C++ 语言和CUDA 软件架构编写，可以在CPU 和 GPU 之间无缝切 换运行，目前支持命令行、Python 语言接口以及Matlab 接口。

由于使用C++ 编写和可以运行在GPU 上 ，Caffe运行速度很快，能够在海量数据上 运行复杂的模型。利用Caffe 进行开发，模型和参数可以通过文本文件进行定义而不是编 写在代码里，在建模方面具有极大的灵活性。

Caffe的网络 (Net) 是由层 (Layer) 组成的有向无环图 (Directed Acyclic Graph, DAG)。 一个典型的网络，开始于数据层，终止于输出层。输出层计算目标任务，比如进 行分类和预测。中间每个网络层，消费一些数据，然后计算结果，输出到下一层。各个网 络层之间的数据传输，通过BLOB 对象进行封装，BLOB 的内容是多维数组。 Caffe 采用 随机梯度下降算法，进行神经网络的训练。

Caffe 可以应用在图像分析、计算机视觉、语音识别、机器翻译、机器人等众多领域。

除 了 TensorFlow 和 Caffe 之外，其他开源的深度学习软件包还有 Theano 和 Torch 等 。

**5** **.** **4** **思** **考** **题**

(1)机器学习与数据挖掘；有监督学习，无监督学习，半监督学习。

(2)决策树的原理及其应用。

(3)K-Means 聚类算法的优点和缺点，如何确定 K-Means 算法的 K。

(4)支持向量机SVM 、 支持向量回归 SVR 的原理及其应用。



(5)EM(Expectation-Maximization) 算法原理及其应用。

(6)协同过滤推荐算法原理与应用。

(7)kNN 算法原理及其应用。

(8)朴素贝叶斯 (Naive Bayes) 算法原理及其应用。

(9)AdaBoost 算法原理及其应用。

(10)线性回归、 Logistic回归原理及其应用。

(11)深度学习、深度学习的应用、AutoEncoder 、RBM 、CNN 、RNN 、DBN 、ISTM。

(12)主流的深度学习软件包。