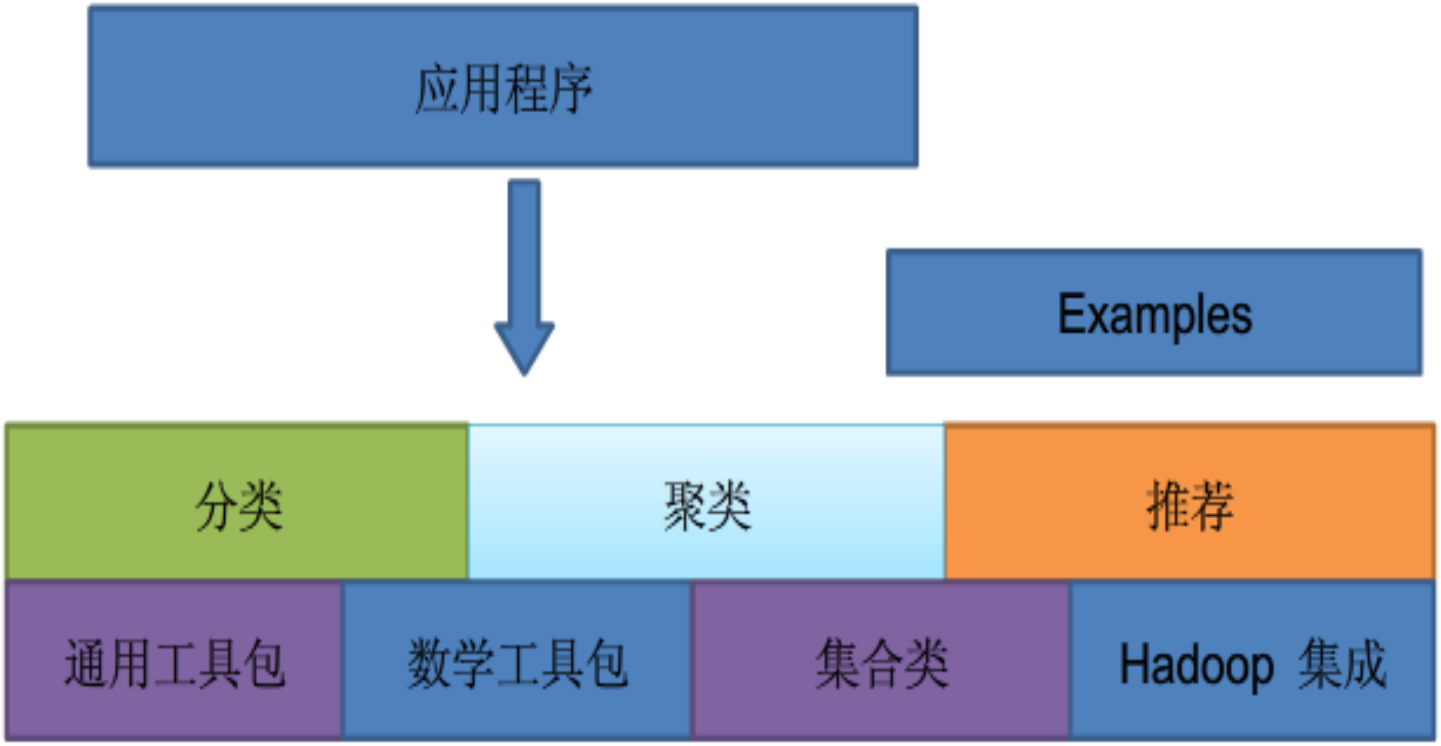
# 数据挖掘组件 Mahout


- Mahout简介
- 什么是机器学习
- Mahout算法介绍
  - 协同筛选
  - 聚类
  - 分类
  - 实战案例

# Mahout简介

Mahout的功能



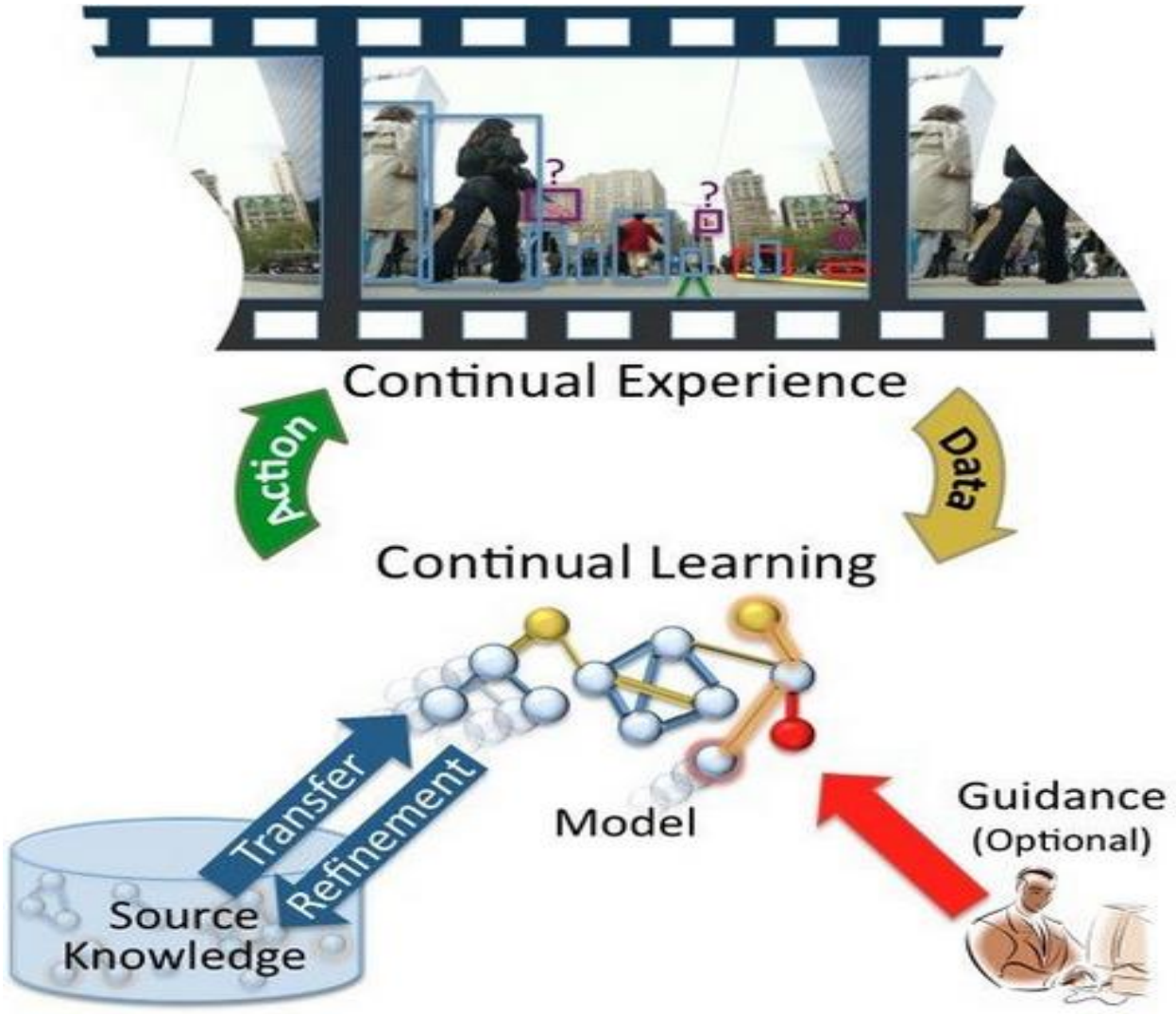
# Mahout简介

- Mahout是一个北印度语的单词，指的是驱使大象的人。
- Mahout项 目开始于2008年，作为Apache Lucene的子项目，Apache Lucene项目是大家熟知的开源搜索引擎。Lucene提供了搜索、文本挖掘和信息检索的高级实现。在计算机科学领域，这些概念和机器学习技术近似，像聚类、分类。所以，Lucene贡献者的一部分机器学习相关工作被剥离进入子项目。不久后，Mahout吸收进“Taste”开源协同过滤的项目。自2010.4月起，Mahout成为Apache的顶级项目。

# Mahout简介

- Mahout的大量工作不只是传统的实现这些算法，也实现将这些算法，让它们工作在hadoop之上。Hadoop的吉祥物是一头大象。

# 机器学习

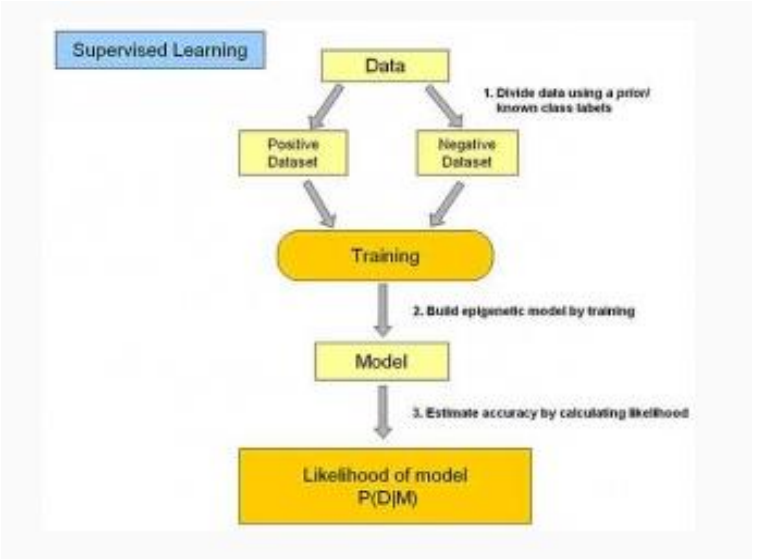

Continual Experience
Action
Data
Continual Learning
Transfer
Refinement
Source Knowledge
Model
Guidance (Optional)

# 机器学习

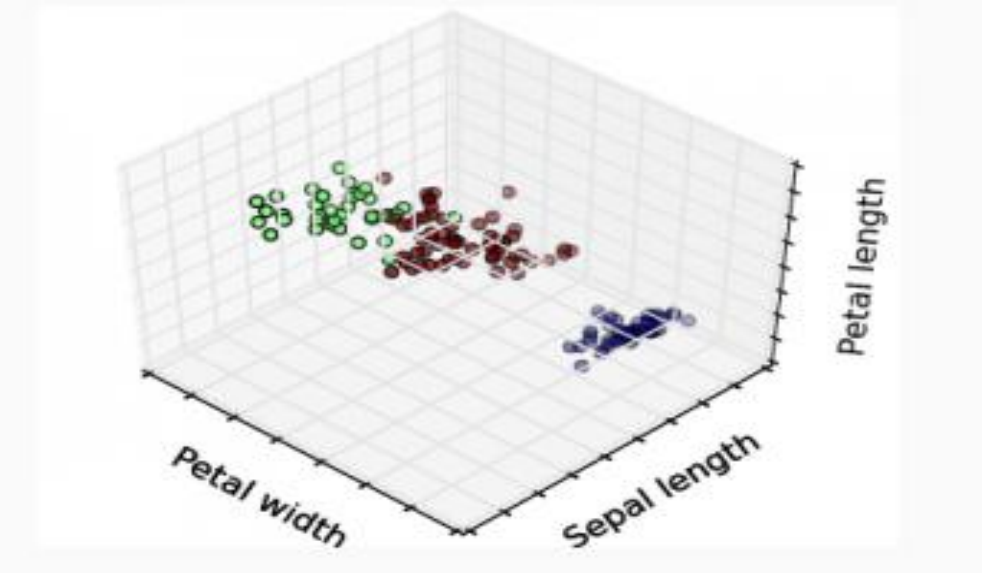
对于机器学习，应该了解的4个学习方式：

- 监督式学习：

  在监督式学习下，输入数据被称为“训练数据”，每组训练数据有一个明确的标识或结果，如对防垃圾邮件系统中“垃圾邮件”“非垃圾邮件”，对手写数字识别中的“1“，”2“，”3“，”4“等。在建立预测模型的时候，监督式学习建立一个学习过程，将预测结果与“训练数据”的实际结果进行比较，不断的调整预测模型，直到模型的预测结果达到一个预期的准确率。监督式学习的常见应用场景如分类问题和回归问题。常见算法有逻辑回归（Logistic Regression）和反向传递神经网络（Back Propagation Neural Network）



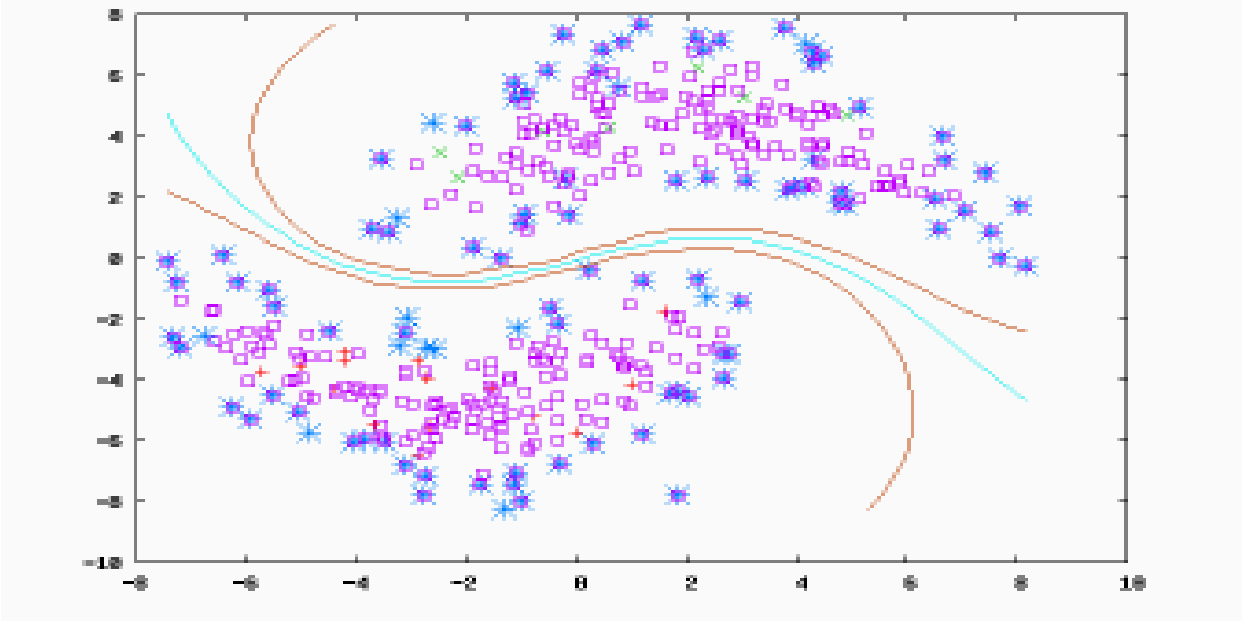
# 机器学习




- 非监督式学习

在非监督式学习中，数据并不被特别标识，学习模型是为了推断出数据的一些内在结构。常见的应用场景包括关联规则的学习以及聚类等。常见算法包括 Apriori 算法以及k-Means 算法。

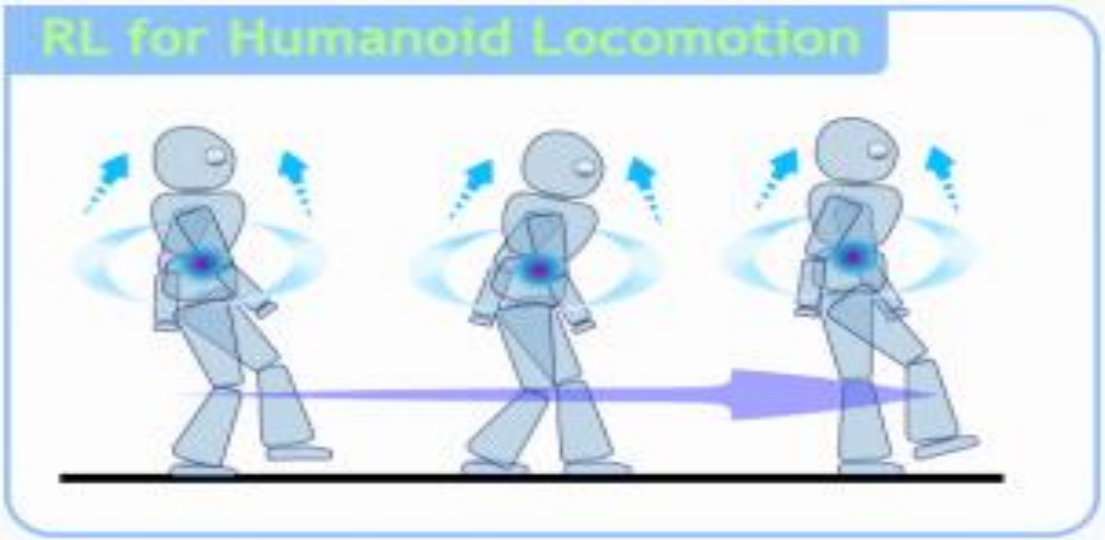
# 机器学习




- 半监督式学习

在此学习方式下，输入数据部分被标识，部分没有被标识，这种学习模型可以用来进行预测，但是模型首先需要学习数据的内在结构以便合理的组织数据来进行预测。应用场景包括分类和回归，算法包括一些对常用监督式学习算法的延伸，这些算法首先试图对未标识数据进行建模，在此基础上再对标识的数据进行预测。如图论推理算法（Graph Inference）或者拉普拉斯支持向量机（Laplacian SVM.）等

# 机器学习



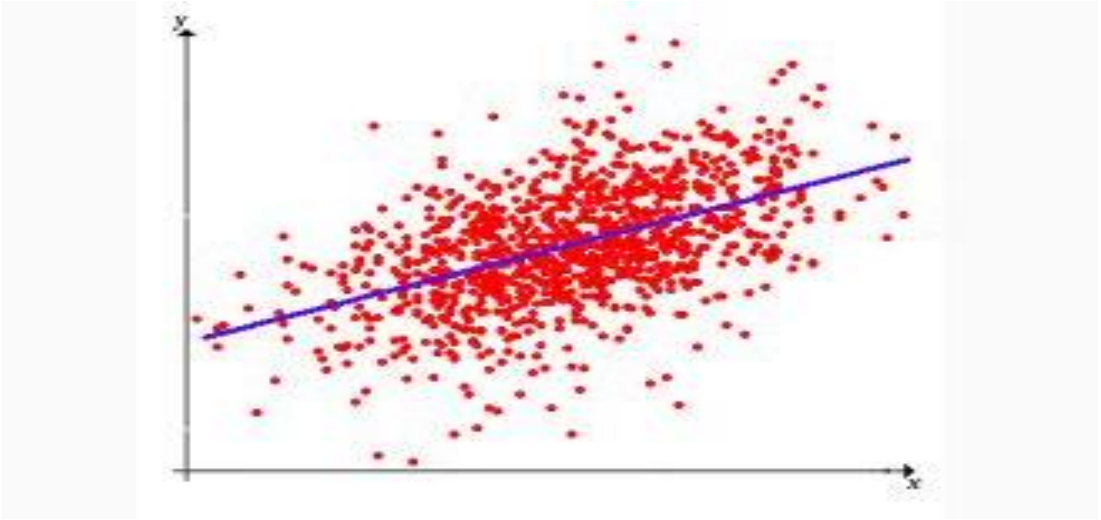

- 强化学习

在这种学习模式下，输入数据作为对模型的反馈，不像监督模型那样，输入数据仅仅是作为一个检查模型对错的方式，在强化学习下，输入数据直接反馈到模型，模型必须对此立刻作出调整。常见的应用场景包括动态系统以及机器人控制等。常见算法包括Q-Learning 以及时间差学习（Temporal difference learning）

# 机器学习

- 根据算法的功能和形式的类似性，我们可以把算法分类，比如说基于树的算法，基于神经网络的算法等等。当然，机器学习的范围非常庞大，有些算法很难明确归类到某一类。而对于有些分类来说，同一分类的算法可以针对不同类型的问题。这里，我们尽量把常用的算法按照最容易理解的方式进行分类。

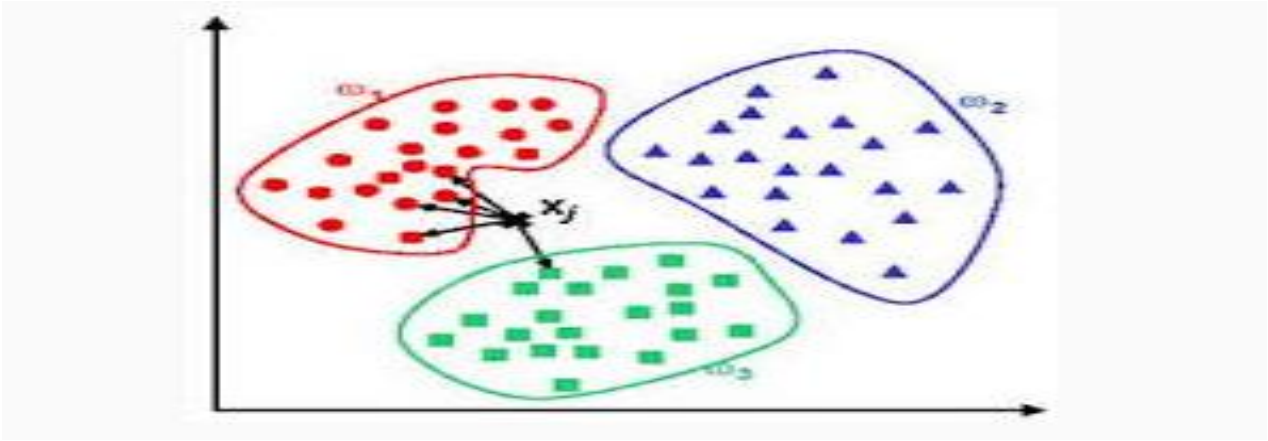
# 机器学习



- 回归算法

回归算法是试图采用对误差的衡量来探索变量之间的关系的一类算法。回归算法是统计机器学习的利器。在机器学习领域，人们说起回归，有时候是指一类问题，有时候是指一类算法，这一点常常会使初学者有所困惑。常见的回归算法包括：最小二乘法（Ordinary Least Square），逻辑回归（Logistic Regression），逐步式回归（Stepwise Regression），多元自适应回归样条（Multivariate Adaptive Regression Splines）以及本地散点平滑估计（Locally Estimated Scatterplot Smoothing）

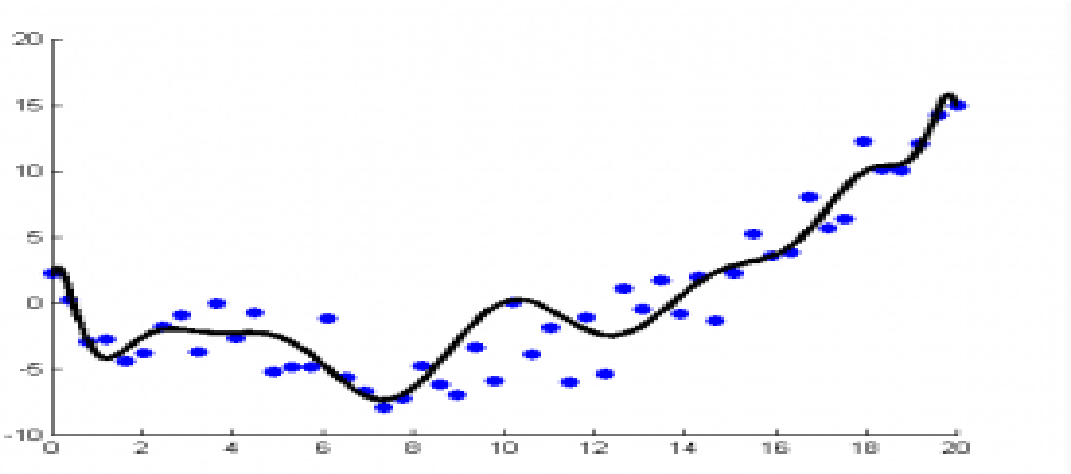
# 机器学习



- 基于实例的算法

基于实例的算法常常用来对决策问题建立模型，这样的模型常常先选取一批样本数据，然后根据某些近似性把新数据与样本数据进行比较。通过这种方式来寻找最佳的匹配。因此，基于实例的算法常常也被称为“赢家通吃”学习或者“基于记忆的学习”。常见的算法包括 k-Nearest Neighbor (KNN)，学习矢量量化（Learning Vector Quantization， LVQ），以及自组织映射算法（Self-Organizing Map ， SOM）

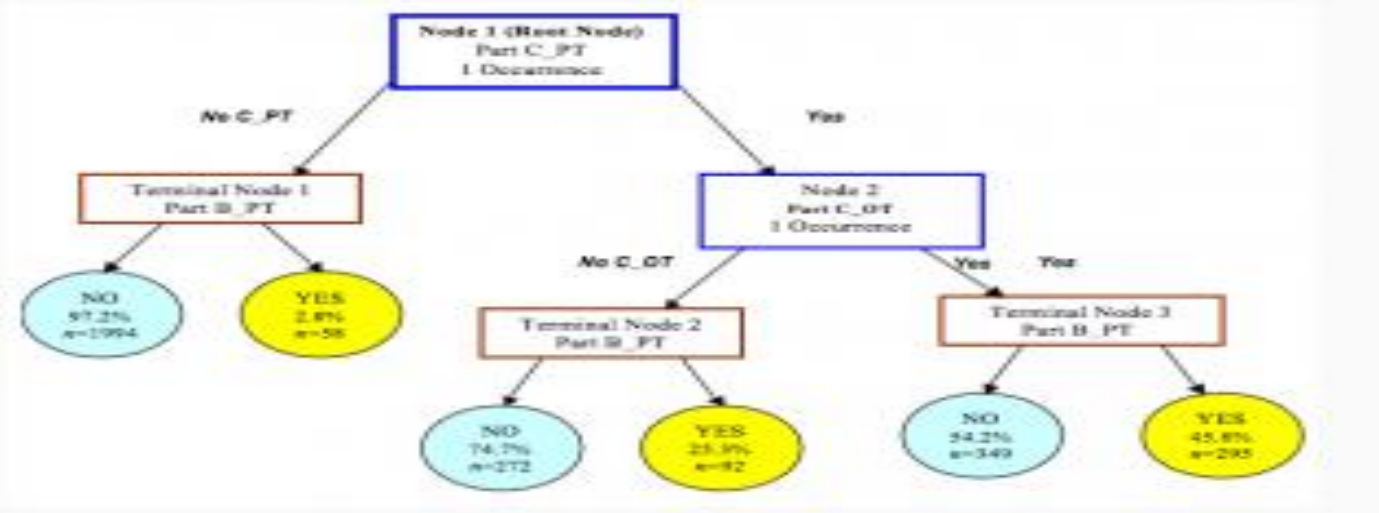
# 机器学习




- 正则化方法

正则化方法是其他算法（通常是回归算法）的延伸，根据算法的复杂度对算法进行调整。正则化方法通常对简单模型予以奖励而对复杂算法予以惩罚。常见的算法包括：Ridge Regression， Least Absolute Shrinkage and Selection Operator（LASSO），以及弹性网络（Elastic Net）。

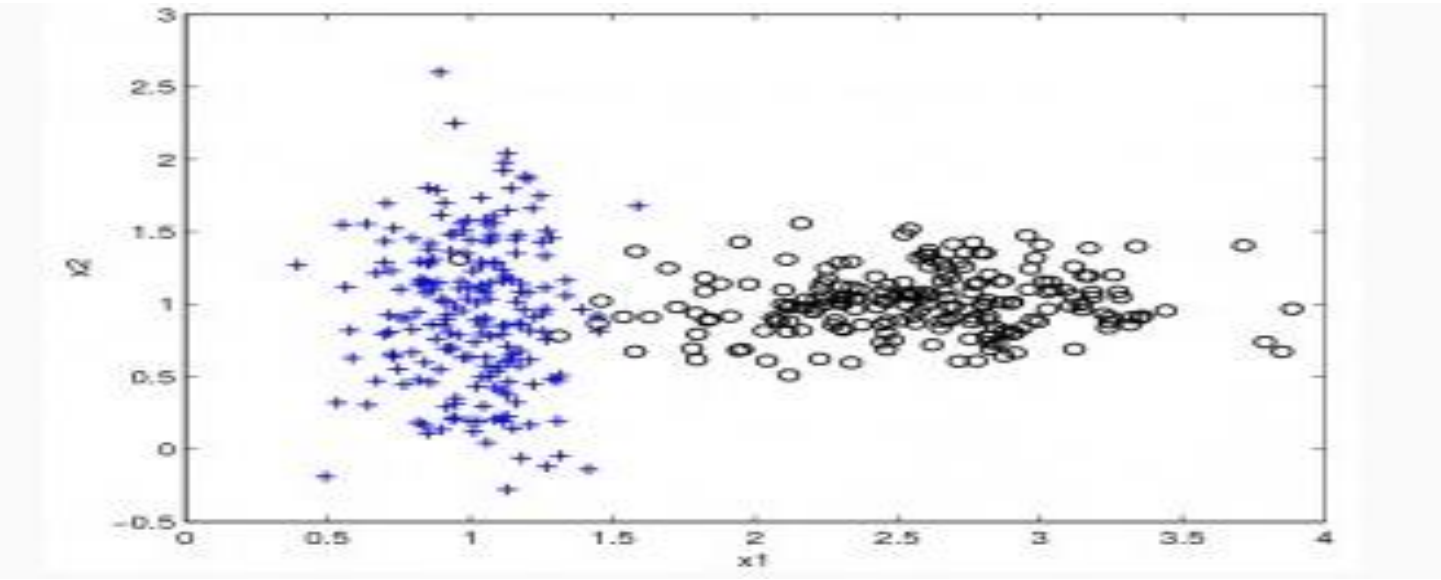
# 机器学习




- 决策树学习

决策树算法根据数据的属性采用树状结构建立决策模型， 决策树模型常常用来解决分类和回归问题。常见的算法包括：分类及回归树（ Classification And Regression Tree， CART ）， ID3 (Iterative Dichotomiser 3)， C4.5， Chi-squared Automatic Interaction Detection (CHAID)， Decision Stump, 随机森林（ Random Forest ），多元自适应回归样条（ MARS ）以及梯度推进机（ Gradient Boosting Machine， GBM ）

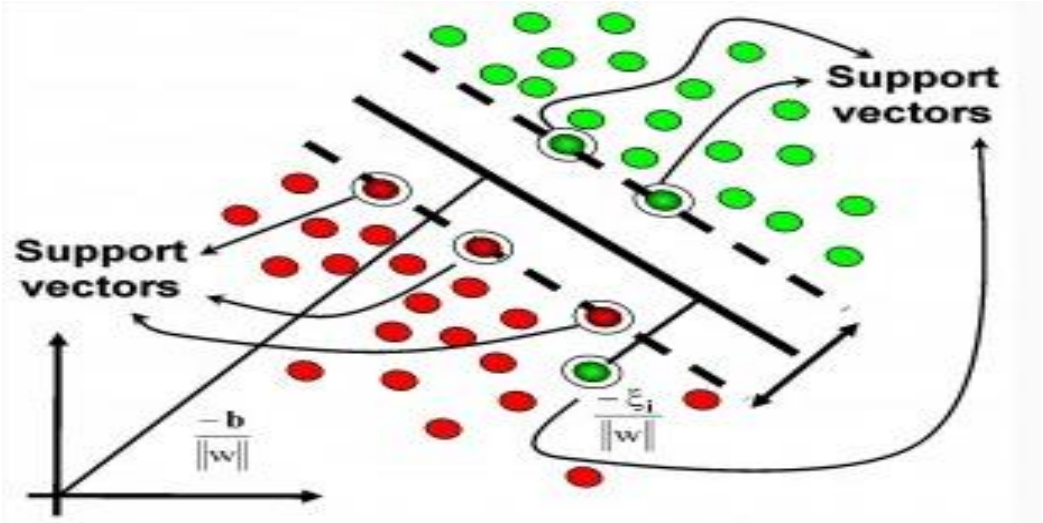
# 机器学习




- 贝叶斯方法

贝叶斯方法算法是基于贝叶斯定理的一类算法，主要用来解决分类和回归问题。常见算法包括：朴素贝叶斯算法，平均单依赖估计（Averaged One-Dependence Estimators， AODE），以及 Bayesian Belief Network（BBN）。

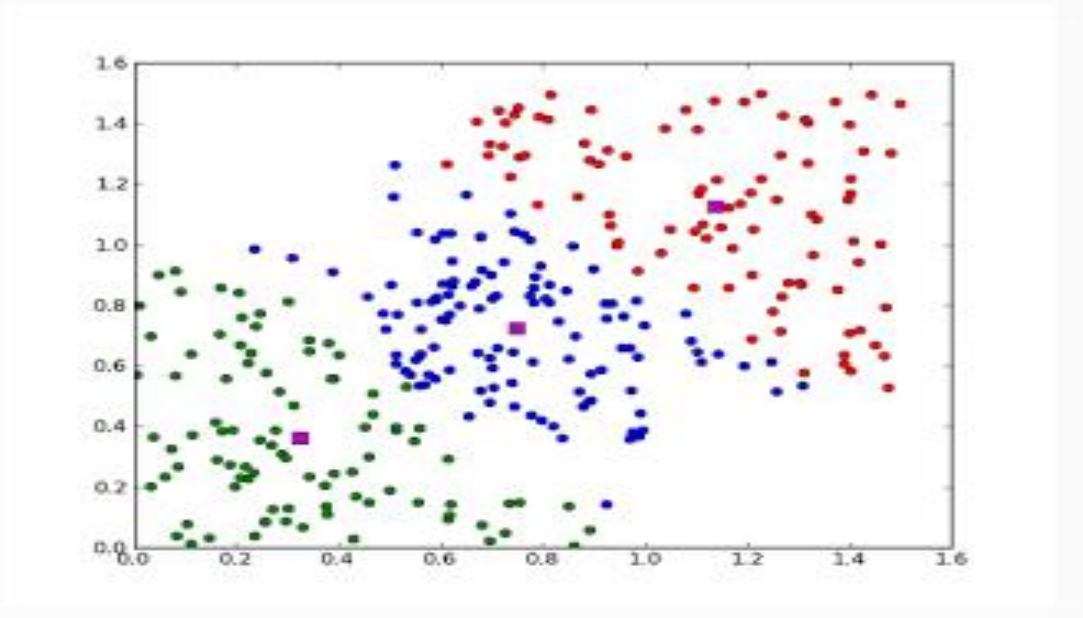
# 机器学习




- 基于核的算法

基于核的算法中最著名的莫过于支持向量机（SVM）了。 基于核的算法把输入数据映射到一个高阶的向量空间， 在这些高阶向量空间里，有些分类或者回归问题能够更容易的解决。 常见的基于核的算法包括：支持向量机（Support Vector Machine， SVM）， 径向基函数（Radial Basis Function ，RBF)， 以及线性判别分析（Linear Discriminate Analysis ，LDA)等。

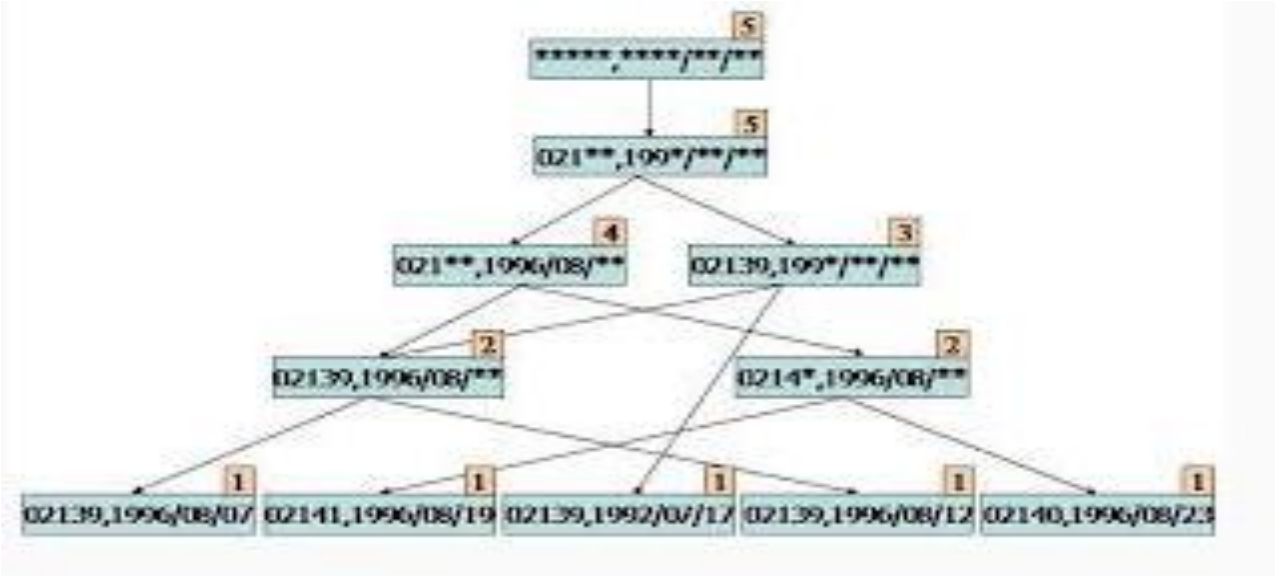
# 机器学习




- 聚类算法

聚类，就像回归一样，有时候人们描述的是一类问题，有时候描述的是一类算法。聚类算法通常按照中心点或者分层的方式对输入数据进行归并。所以的聚类算法都试图找到数据的内在结构，以便按照最大的共同点将数据进行归类。常见的聚类算法包括 k-Means 算法以及期望最大化算法（Expectation Maximization， EM）。

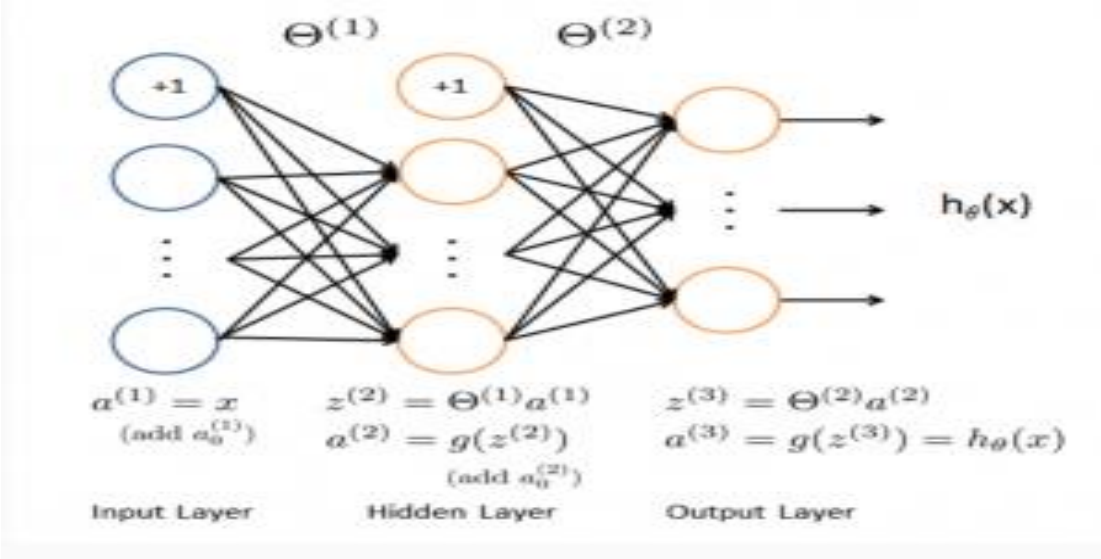
# 机器学习




- 关联规则学习

关联规则学习通过寻找最能够解释数据变量之间关系的规则，来找出大量多元数据集中有用的关联规则。常见算法包括 Apriori 算法和 Eclat 算法等。

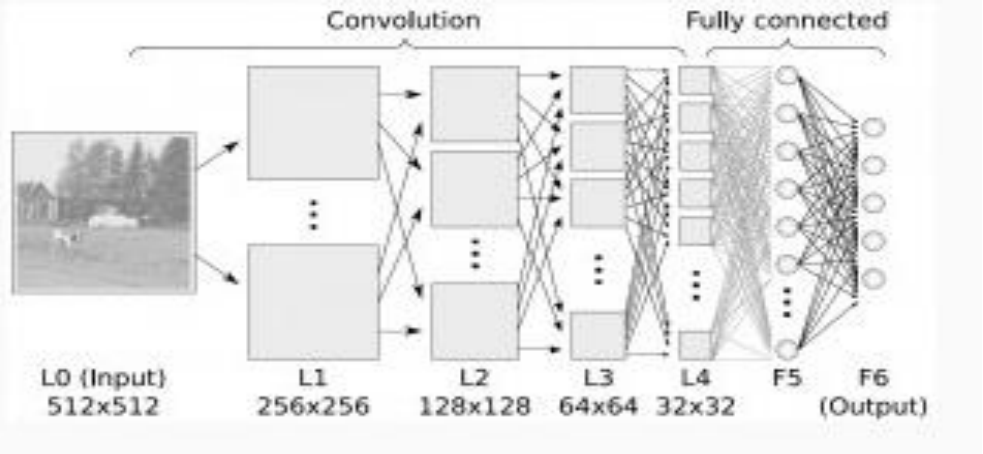
# 机器学习




- 人工神经网络

人工神经网络算法模拟生物神经网络，是一类模式匹配算法。通常用于解决分类和回归问题。人工神经网络是机器学习的一个庞大的分支，有几百种不同的算法。（其中深度学习就是其中的一类算法，我们会单独讨论），重要的人工神经网络算法包括：感知器神经网络（Perceptron Neural Network），反向传递（Back Propagation），Hopfield 网络，自组织映射（Self-Organizing Map, SOM）。学习矢量量化（Learning Vector Quantization， LVQ）

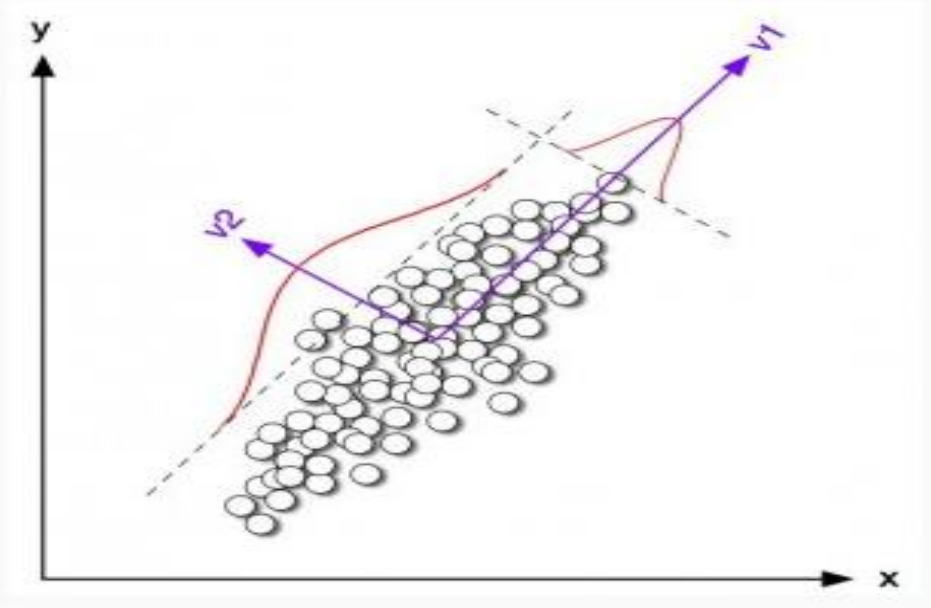
# 机器学习




- 深度学习

深度学习算法是对人工神经网络的发展。 在近期赢得了很多关注，特别是 百度也开始发力深度学习后， 更是在国内引起了很多关注。 在计算能力变得日益廉价的今天，深度学习试图建立大得多也复杂得多的神经网络。很多深度学习的算法是半监督式学习算法，用来处理存在少量未标识数据的大数据集。常见的深度学习算法包括：受限波尔兹曼机（Restricted Boltzmann Machine， RBN）， Deep Belief Networks（DBN），卷积网络（Convolutional Network）， 堆栈式自动编码器（Stacked Auto-encoders）。

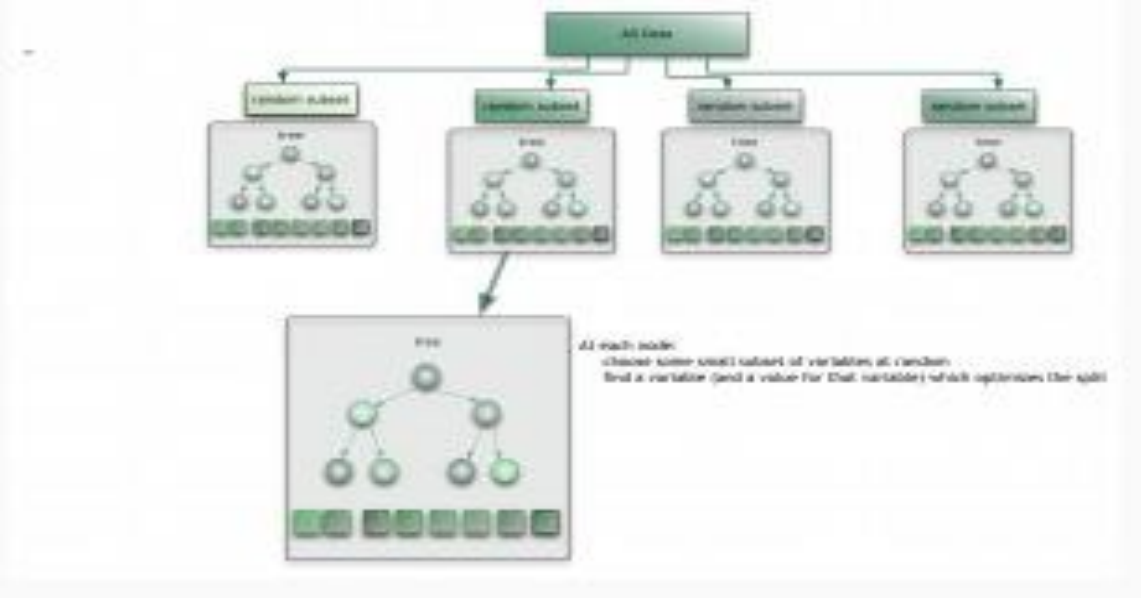
# 机器学习




- 降低维度算法

像聚类算法一样，降低维度算法试图分析数据的内在结构，不过降低维度算法是以非监督学习的方式试图利用较少的信息来归纳或者解释数据。这类算法可以用于高维数据的可视化或者用来简化数据以便监督式学习使用。常见的算法包括：主成份分析（Principle Component Analysis， PCA），偏最小二乘回归（Partial Least Square Regression，PLS）， Sammon 映射，多维尺度（Multi-Dimensional Scaling, MDS）， 投影追踪（Projection Pursuit）等。

# 机器学习



- 集成算法

集成算法用一些相对较弱的学习模型独立地就同样的样本进行训练，然后把结果整合起来进行整体预测。集成算法的主要难点在于究竟集成哪些独立的较弱的学习模型以及如何把学习结果整合起来。这是一类非常强大的算法，同时也非常流行。常见的算法包括：Boosting， Bootstrapped Aggregation（Bagging），AdaBoost，堆叠泛化（Stacked Generalization， Blending），梯度推进机（Gradient Boosting Machine, GBM），随机森林（Random Forest）。

# 机器学习常用算法

数据挖掘模型

Data Mining Model

- 监督模型、预测模型 Supervised Model Predictive Model
  - 神经网络 Neural Networks
  - 决策树 C5.0
  - 决策树 C&RT(CART)
  - 回归 Regression
  - 逻辑回归 Logistic regression（分类变量预测）
- 无监督模型 Unsupervised Model
  - 聚类分析 Clustering
    - 神经网络算法 Kohonen
    - 快速聚类 K-means
    - 二阶聚类 Two-Step
  - 关联分析 Associations
    - Apriori算法
    - 多维关联 GRI
    - 时序关联 Sequence
- 数据降维 Data Reduction
  - 主成分分析 PCA
  - 因子分析 Factor

# Mahout算法介绍

- Apache Mahout 项目旨在帮助开发人员更加方便快捷地创建智能应用程序。程序员无需关注底层的算法实现，如果对算法有研究，Mahout也提供对应的接口来修改对应的算法逻辑来匹配业务需求。
- Mahout 算法所处理的场景，经常是伴随着海量的用户使用数据的情况。通过将 Mahout 算法构建于 MapReduce 框架之上，将算法的输入、输出和中间结果构建于 HDFS 分布式文件系统之上，使得 Mahout 具有高吞吐、高并发、高可靠性的特点。最终，使业务系统可以高效快速地得到分析结果。

# Mahout算法介绍-推荐引擎算法

推荐引擎算法：

- Taste是 Apache Mahout 提供的一个个性化推荐引擎的高效实现，该引擎基于java实现，可扩展性强，同时在mahout中对一些推荐算法进行了MapReduce编程模式转化，从而可以利用hadoop的分布式架构，提高推荐算法的性能

# Mahout算法介绍-推荐引擎算法

推荐引擎算法：

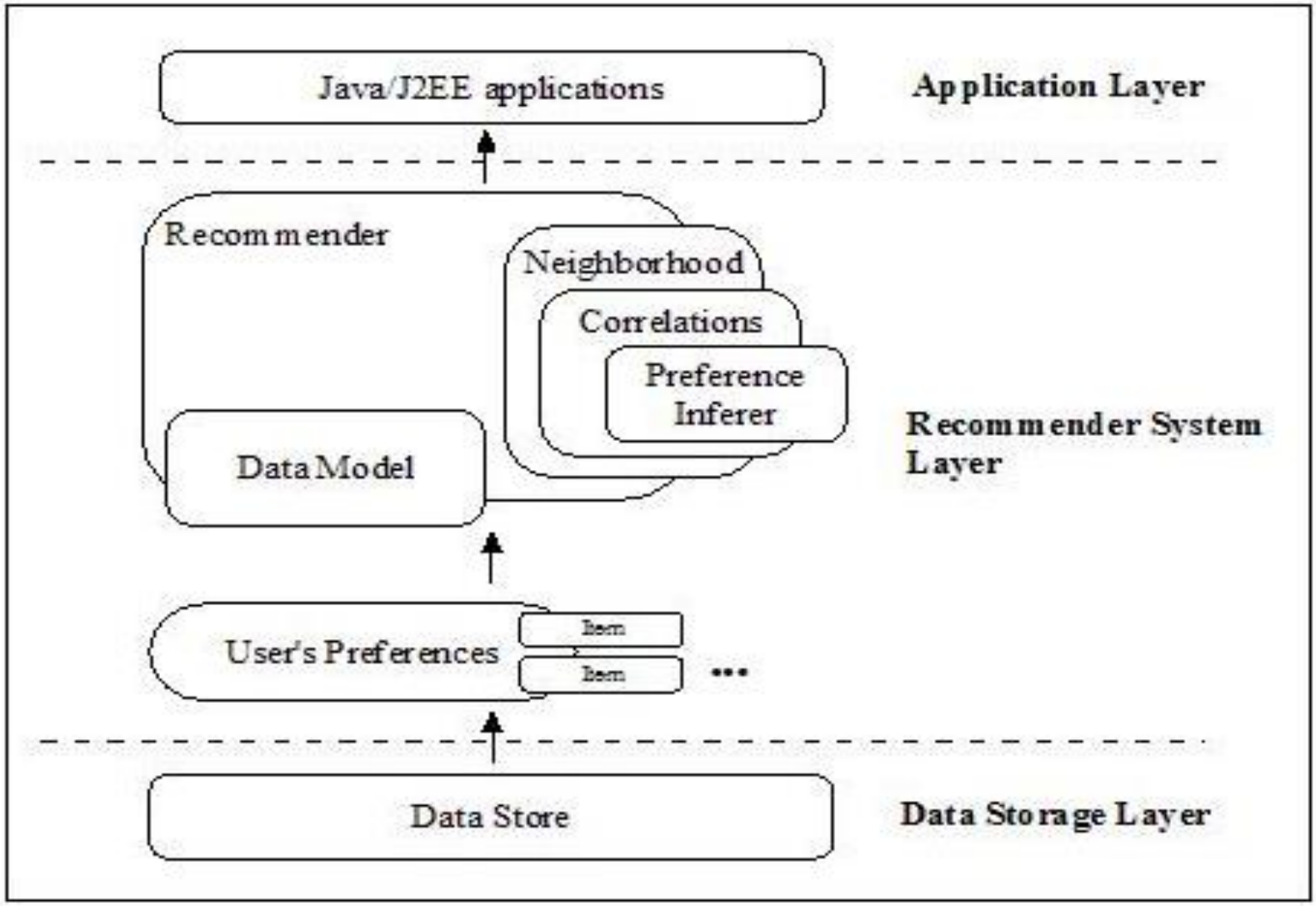
- 在Mahout0.5版本中的Taste， 实现了多种推荐算法，其中有最基本的基于用户的和基于内容的推荐算法，也有比较高效的SlopeOne算法，以及处于研究阶段的基于SVD和线性插值的算法，同时Taste还提供了扩展接口，用于定制化开发基于内容或基于模型的个性化推荐算法。

# Mahout算法介绍-推荐引擎算法

推荐引擎算法：

- Taste 不仅仅适用于 Java 应用程序，还可以作为内部服务器的一个组件以 HTTP 和 Web Service 的形式向外界提供推荐的逻辑。Taste 的设计使它能满足企业对推荐引擎在性能、灵活性和可扩展性等方面的要求。

# Mahout算法介绍-推荐引擎算法



# Mahout算法介绍-推荐引擎算法

●DataModel：DataModel是用户喜好信息的抽象接口，它的具体实现支持从指定类型的数据源抽取用户喜好信息。在Mahout0.5中，Taste 提供 JDBCDataModel 和 FileDataModel两种类的实现，

●分别支持从数据库和文件文件系统中读取用户的喜好信息。对于数据库的读取支持，在Mahout 0.5中只提供了对MySQL和PostgreSQL的支持，如果数据存储在其他数据库，或者是把数据导入到这两个数据库中，或者是自行编程实现相应的类。

# Mahout算法介绍-推荐引擎算法

- UserSimilarit和ItemSimilarity：前者用于定义两个用户间的相似度，后者用于定义两个项目之间的相似度。Mahout支持大部分驻留的相似度或相关度计算方法，针对不同的数据源，需要合理选择相似度计算方法
- redis
- memcached

# Mahout算法介绍-推荐引擎算法

- UserNeighborhood：在基于用户的推荐方法中，推荐的内容是基于找到与当前用户喜好相似的“邻居用户”的方式产生的，该组件就是用来定义与目标用户相邻的“邻居用户”。所以，该组件只有在基于用户的推荐算法中才会被使用。

# Mahout算法介绍-推荐引擎算法

- Recommender：Recommender是推荐引擎的抽象接口，Taste 中的核心组件。利用该组件就可以为指定用户生成项目推荐列表。

# Mahout算法介绍-聚类算法

**聚类算法:**

- 聚类 (Clustering) 就是将数据对象分组成为多个类或者簇 (Cluster)，它的目标是：在同一个簇中的对象之间具有较高的相似度，而不同簇中的对象差别较大。所以，在很多应用中，一个簇中的数据对象可以被作为一个整体来对待，从而减少计算量或者提高计算质量。

# Mahout算法介绍-聚类算法

- 聚类同时也在 Web 应用中起到越来越重要的作用。最被广泛使用的既是对 Web 上的文档进行分类，组织信息的发布，给用户一个有效分类的内容浏览系统（门户网站），同时可以加入时间因素，进而发现各个类内容的信息发展

# Mahout算法介绍-聚类算法

- 最近被大家关注的主题和话题，或者分析一段时间内人们对什么样的内容比较感兴趣，这些有趣的应用都得建立在聚类的基础之上。作为一个数据挖掘的功能，聚类分析能作为独立的工具来获得数据分布的情况，观察每个簇的特点，集中对特定的某些簇做进一步的分析，此外，聚类分析还可以作为其他算法的预处理步骤，简化计算量，提高分析效率，这也是我们在这里介绍聚类分析的目的。

# Mahout算法介绍-聚类算法

| 算法 | 内存实现 | Map/Reduce 实现 | 簇个数是确定的 | 簇是否允许重叠 |
|---|---|---|---|---|
| K 均值 | KMeansClusterer | KMeansDriver | Y | N |
| Canopy | CanopyClusterer | CanopyDriver | N | N |
| 模糊 K 均值 | FuzzyKMeansClusterer | FuzzyKMeansDriver | Y | Y |
| 狄利克雷 | DirichletClusterer | DirichletDriver | N | Y |

上图为Mahout内置实现的聚类算法，下面会介绍。

# Mahout算法介绍-聚类算法

- K 均值是典型的基于距离的排他的划分方法：给定一个 n 个对象的数据集，它可以构建数据的 k 个划分，每个划分就是一个聚类，并且 k<=n，同时还需要满足两个要求：
  每个组至少包含一个对象
  每个对象必须属于且仅属于一个组。
- Canopy 聚类算法的基本原则是首先应用成本低的近似的距离计算方法高效的将数据分为多个组，这里称为一个 Canopy，我们姑且将它翻译为“华盖”，Canopy 之间可以有重叠的部分；然后采用严格的距离计算方式准确的计算在同一 Canopy 中的点，将他们分配与最合适的簇中。Canopy 聚类算法经常用于 K 均值聚类算法的预处理，用来找合适的 k 值和簇中心。

# Mahout算法介绍-聚类算法

- 模糊 K 均值聚类算法是 K 均值聚类的扩展，与 K 均值聚类原理类似，模糊 K 均值也是在待聚类对象向量集合上循环，但是它并不是将向量分配给距离最近的簇，而是计算向量与各个簇的相关性（Association）。假设有一个向量 v，有 k 个簇，v 到 k 个簇中心的距离分别是 d1，d2... dk，那么 V 到第一个簇的相关性 u1可以通过下面的算式计算：

# Mahout算法介绍-聚类算法

$$u_1 = \frac{1}{\left(\frac{d_1}{d_1}\right)^{\frac{2}{m-1}} + \left(\frac{d_1}{d_2}\right)^{\frac{2}{m-1}} + \ldots + \left(\frac{d_1}{d_k}\right)^{\frac{2}{m-1}}}$$

●狄利克雷聚类算法是基于概率分布模型的聚类算法，其需要预先定义一个分布模型，然后按照模型对数据进行分类，将不同的对象加入一个模型，模型会增长或者收缩；每一轮过后需要对模型的各个参数进行重新计算，同时估计对象属于这个模型的概率。所以说，基于模型的聚类算法的核心是定义模型，对于一个聚类问题，模型定义的优劣直接影响了聚类的结果。

# Mahout算法介绍-实战案例

前面介绍了Mahout内置的一些实现算法，这里举例说明算法的适用场景。

●推荐引擎是目前我们使用的机器学习技术中最容易识别的。你可能已经见过相关的服务或网页，基于历史行为推荐书、电影、文档。他们尝试推论出用户偏好，并标记出用户不知晓的、感兴趣的item:

① Amazon.com可能是最出名的使用推荐系统商务网站。基于交易和网页活性，Amazon推荐给用户可能感兴趣的书籍和其他item。请参见图1.2（见附件）

② Netflix类似于推荐用户感兴趣的DVDs，并且为研究者提供百万大奖去提升推荐质量。

# Mahout算法介绍-实战案例

① 约会网站像Líbímseti将一部分用户推荐给其他用户

② 社交网络网站像Facebook用推荐技术的变形来为用户识别最可能成为一个尚未建立联系的朋友。

③ 对于Amazon和示例其他网站，通过这种聪明的交叉销售，推荐系统确实有具体的经济价值，同一家公司的报告指出推荐产品给用户能够带来8-12%的销售增长。

# Mahout算法介绍-实战案例

●聚类技术尝试去将大量的拥有相同相似度的事物聚集到不同的类中。聚类是在海量或者难于理解的数据集里发现层次和顺序，展现兴趣模式，或使得数据集容易被理解。

①Google News据为了根据具备逻辑性的故事聚集展示新闻，而不是所有文章的行列表，使用新闻文章的Topic聚集新闻。

②搜索引擎像Clusty基于相同的原因聚集搜索结果。

③使用聚类技术，基于消费者属性，收入、位置、购买习惯，可将不用用户分到不用的类中。

# Mahout算法介绍-实战案例

●分类技术用于决定一个事物是不是属于一种类型、类目，或者该事物是不是含有某些属性。同样地，分类无处不在，尽管更多的时候隐于幕后。

●这些系统通过评估item的很多实例来学习，以推导出分类规则。这个平常的想法可以找到很多应用：

①Yahoo! Mail决定接收的信息是不是垃圾邮件，基于先前邮件和用户的垃圾邮件报告，以及邮件的特性。一些信息被分类为垃圾邮件，

②Picasa (http://picasa.google.com/)和其他的照片管理应用可以判断一张照片中是否含有人脸。

# Mahout算法介绍-实战案例

①光学字符识别软件通过将小区域作为独立字符来分类，将扫描文本的若干小区域归类到独立的字符上。

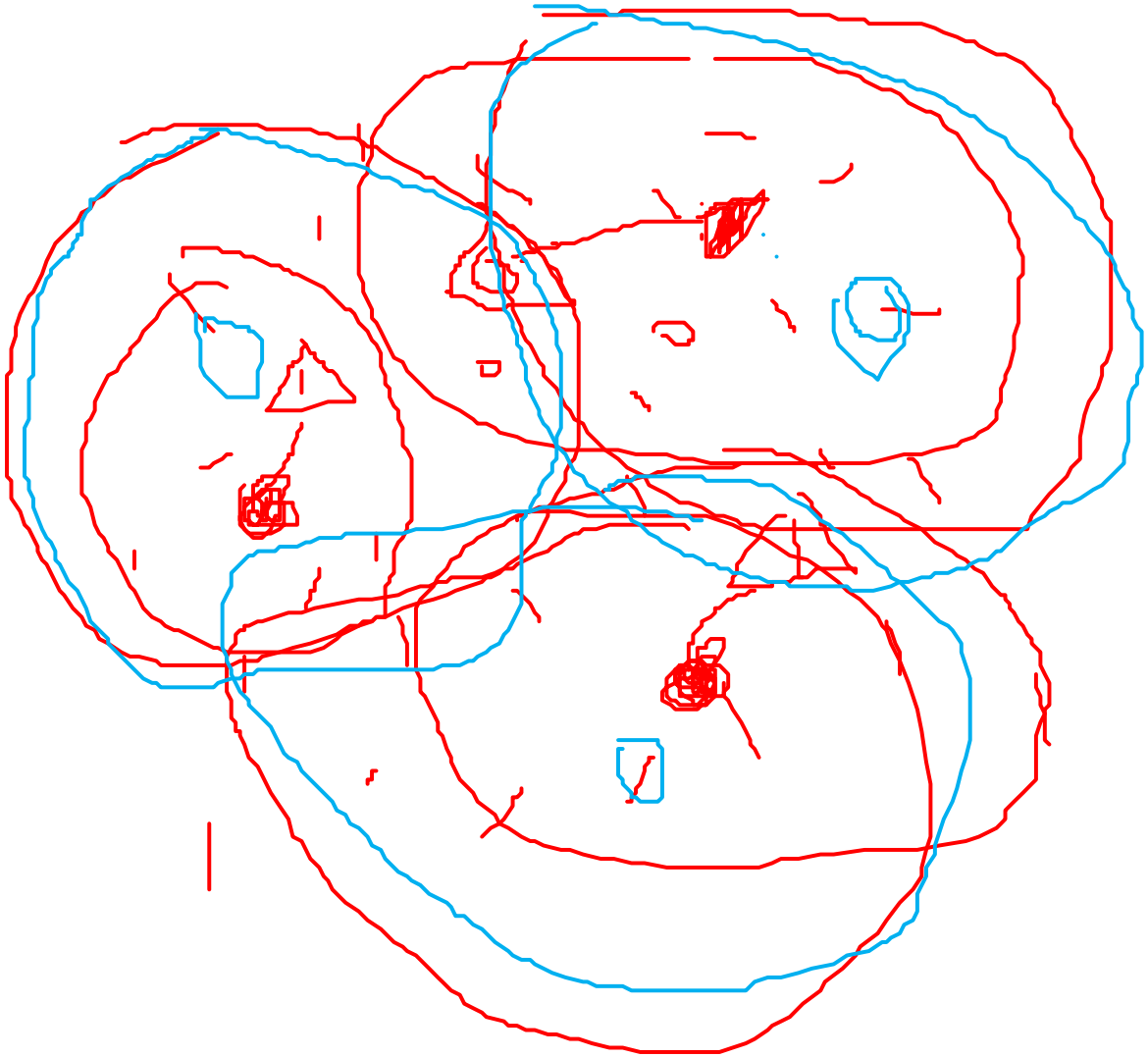
②在iTunes中Apple’s Genius feature使用分类将歌曲划分到不同的潜在播放列表。

分类有助于判断一个新进入事物是否匹配先前发现的模式，也常用于分类行为或者模式。分类也可用来检测可疑的网络活动或欺诈。也可用于根据用户发信息判定表示失望或者满意。

# Mahout总结

本篇PPT主要介绍:

- Mahout的发展过程
- 机器学习的基本概念和经典算法
- Mahout内置的算法实现
- Mahout在实际中的使用场景


k=3