Mahout数据挖掘工具包介绍

讲师：董西成
博客：dongxicheng.org
微信号：hadoop-123
（二维码见右）

Open Passion Value

目录

传统数据挖掘/机器学习库存在的问题

- 缺少一个活跃的技术社区
- 扩展性差
- 文档化差，缺少实例
- 不开源，商业化库
- 通常由研究机构开发
- 实施性差

Apache Mahout优点

- 技术社区活跃
- 扩展性好
- 文档化好，实例丰富
- 100%源代码开源
- 易于使用

Apache Mahout是什么

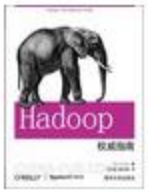
- 基于MapReduce开发的数据挖掘/机器学习库
- 良好的扩展性和容错性
  - ✓ 充分利用了MapReduce和HDFS的扩展性和容错性
- 属于Hadoop生态系统重要组成部分
  - ✓ Apache Software License 2
- 实现了大部分常用的数据挖掘算法
  - ✓ 聚类算法
  - ✓ 分类算法
  - ✓ 推荐算法

Apache Mahout应用场景-物品/人推荐

喜欢读"Hadoop技术内幕"的人也喜欢

Java并发编程实战

Linux内核设计与实现

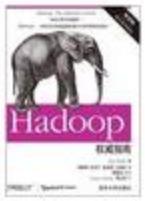


Hadoop权威指南(中文版)

大数据

代码整洁之道

深入理解计算机系统（原书第2版）



Hadoop权威指南

鸟哥的Linux私房菜.基础学习篇（第...

深入理解Java拟机

可能感兴趣的人 换一换


李航博士 V
+关注

华为技术有限公司 诺亚方...
关注数据科学家联...的人还关注


南非蜘蛛 V
+关注

中国最大开源社区Chinauni...
程开源、IT人等10个好友也关注


Burberry V
+关注

PEOPLE YOU MAY KNOW

Jun Gong, Software Engineer at EMC
Connect

as lily, Student at Beijing Institute of Technology
Connect

andy deng, ASIC Engineer at VIA Technologies, Inc.
Connect

See more »

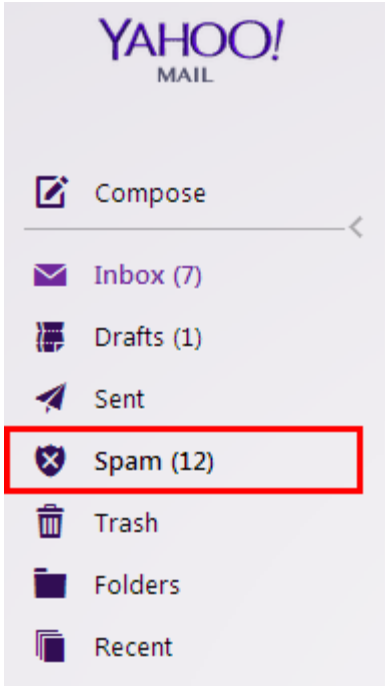
You May Be Interested In

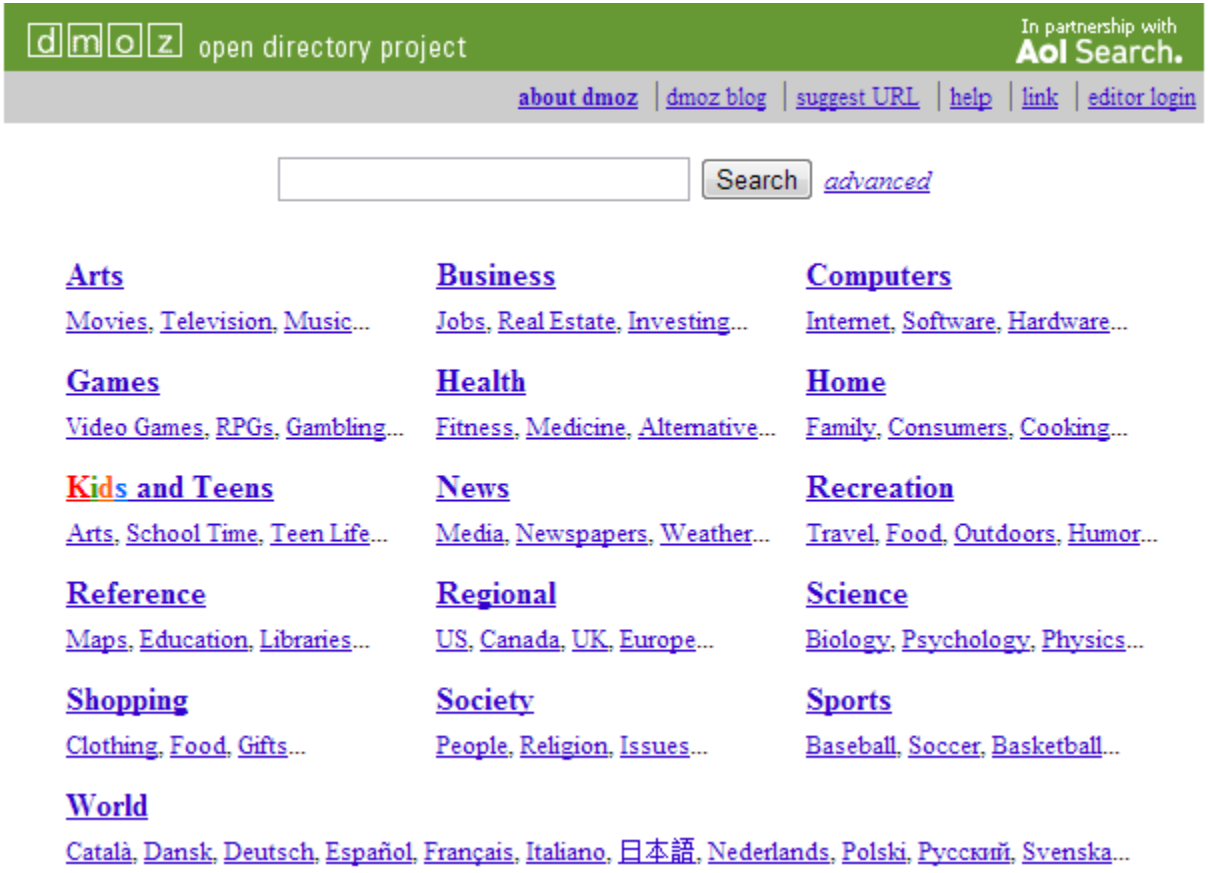
Study Business Online
Start your Australian Diploma in Business or Project Management Today

rofit Certificate
ur passion become your
with just four online courses

USA and UK VPN - Unblock
Use StrongVPN & have a USA or UK IP address anywhere. Remove blocks now

Apache Mahout应用场景—分类





Apache Mahout应用场景—聚类

首页 | 国内 | 国际 | 军事 | 财经 | 互联网 | 房产 | 汽车 | 体育 | 娱乐 | 游戏 | 教育 | 女人 | 科技 | 社会

互联网新闻 人物动态 | 公司动态 | 搜索引擎 | 电子商务 | 网络游戏 | 传闻爆料 | 互联网评论

京东二线城市试验便利店O2O：样本还是特例？ 10:31 49条相关>>

- 眼镜行业走到十字路口 O2O将成竞争主战场
- 京东和唐久试水O2O 利益冲突依然是难题 (1)

- 马云谈淘宝"双12"：为实现"双百万" 11-25 16:02 81条相关>>
- 10个人，100万美元，90天：打造一家挑战亚马逊的.. 11-25 15:43 3条相关>>
- 如何让你的客户更忠诚？你需要一个计划 11-25 08:18 8条相关>>
- 一场无声的战争 阿里腾讯攻防战全面升级 11-25 08:33 47条相关>>
- 美国感恩节忙排队：网购热度远不及中国 11-24 11:24 52条相关>>

- 小米试水微信网购 15万台小米手机3开放购买 11-22 14:03 90条相关>>
- 【钛晨报】床单毛巾洗衣粉，请认准亚马逊牌？ 11-21 07:03
- 今日嗅评：快给路由器升级，再简单些、好玩些、.. 11-20 18:46
- 天灏资本维持当当网持有评级 11-19 21:54

- 移动支付还未普及，何来爆发？ 08:14 35条相关>>
- 京东为什么要做专属DSP JD商务舱？ 11-18 19:58 64条相关>>
- 短评：腾讯财付通小贷的价值 11-18 23:20 30条相关>>
- 谁是金融业门口的野蛮人？ 11-20 15:31
- 谁说亚马逊是洗劫出版业的"罪魁祸首"？ 07:43 2条相关>>
- "二马"战指尖 谁威胁了马云？ 11-23 03:48 90条相关>>

目录

Mahout提供的算法

See http://cwiki.apache.org/confluence/display/MAHOUT/Algorithms

Mahout中的分类算法

- **Logistic Regression**
  - ✓ 逻辑回归
- **Bayesian**
  - ✓ 贝叶斯分类算法
- **Support Vector Machines**
  - ✓ 支持向量机
- **Perceptron and Winnow**
  - ✓ 感知器算法
- **Neural Network**
  - ✓ 神经网络
- **Random Forests**
  - ✓ 随机森林
- **Restricted Boltzmann Machines**
  - ✓ 有限波尔兹曼机
- **Online Passive Aggressive**
- **Boosting**
- **Hidden Markov Models**
  - ✓ 隐式马尔科夫链

Mahout中的聚类算法

- Canopy Clustering
- K-Means
- Fuzzy K-Means
  - ✓ 模糊K-Means
- Expectation Maximization
  - ✓ EM算法
- Mean Shift
  - ✓ 均值漂移
- Hierarchical Clustering
  - ✓ 层次聚类
- Dirichlet Process Clustering
  - ✓ 狄里克雷过程聚类
- Latent Dirichlet Allocation
  - ✓ LDA
- Spectral Clustering
  - ✓ 谱聚类
- Minhash Clustering
- Top Down Clustering
  - ✓ 自上而下聚类

Mahout中的其他算法

- **Pattern Mining**
  - Parallel FP Growth
- **Regression**
  - Locally Weighted Linear Regression
- **Dimension Reduction**
  - SVD
  - Stochastic SVD with PCA
  - PCA
  - Independent Component Analysis
  - Gaussian Discriminative Analysis
- **Evolution Algorithms**
  - Genetic Algorithms
- **Recommenders**
  - Non-distributed recommenders ("Taste")
  - Distributed Item-Based Collaboration Filtering
  - Collaboration Filtering using a parallel matrix factorization
  - Slope One

Mahout中的其他算法

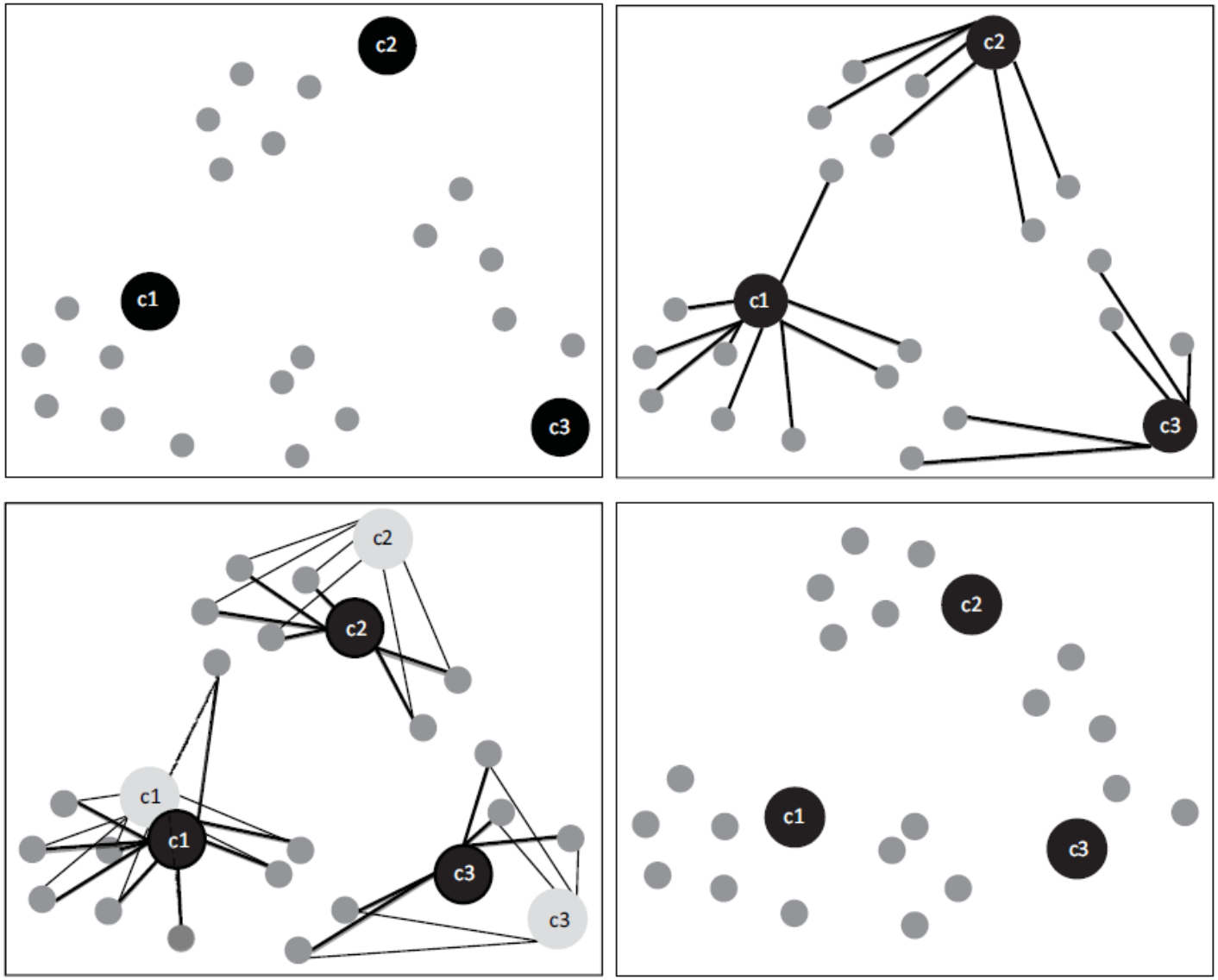
- **Vector Similarity**
  - RowSimiliarityJob (MR)
  - VectorDistanceJob (MR)
- **Other**
  - Collocations
- **Non-MapReduce algorithms**

目录

聚类算法介绍

- 将类似的对象划分成多个类的过程
- “物以类聚，人以群分”
- 以k-means聚类算法为例介绍
  - ✓ 给定聚类个数k
  - ✓ 按照数据特征，将其分为k个类别

聚类算法-K-means



聚类算法-K-means

【K-Means计算步骤】

步骤1：提取feature

步骤2：将feature向量化

步骤3：利用K-means算法实现聚类

【一个实例】

有1TB的新闻网页数据，如何使用K-mean对其分成若干类？

步骤：提取feature

步骤1：提取新闻正文，并分词

步骤2：对词编码，从1开始….

步骤3：每个新闻文档是一个向量，每个元素为0或者1

比如：

	1 (i)	2 (boy)	3 (car)	4 (girl)	5 (run)	6 (fly)	7 (fid)	8 (app)	9 (or)
doc1	0	0	1	0	1	0	0	0	1
doc2	1	1	0	0	0	1	0	1	0
doc3	0	1	0	1	1	1	0	0	0

步骤2：将feature向量化

- 向量表示形式
  - ✓ DenseVector：元素数目即为feature总数，不管是0还是1
  - ✓ RandomAccessSparseVector：采用hashmap保存，更省空间
  - ✓ SequentialAccessSparseVector：经优化，便于随机存取
- 向量转化方式
  - ✓ 通过Lucene相关库
    - bin/mahout lucenevector …
  - ✓ 通过mahout自带工具
    - bin/mahout seqdirectory …
    - bin/mahout seq2sparse …
  - ✓ 自己编写程序
- 将数据保存成SequenceFile格式

步骤3：利用K-means算法实现聚类

```
bin/mahout kmeans \
    -i <input vectors directory> \
    -c <input clusters directory> \
    -o <output working directory> \
    -k <optional number of initial clusters to sample from input vectors> \
    -dm <DistanceMeasure> \
    -x <maximum number of iterations> \
    -cd <optional convergence delta. Default is 0.5> \
    -ow <overwrite output directory if present>
    -cl <run input vector clustering after computing Canopies>
    -xm <execution method: sequential or mapreduce>
```

- **-i：指定待分类数据集所在路径，数据格式需为SequenceFile(WritableComparable, VectorWritable)，其中key不会被使用（被忽略）；**
- **-c：指定初始类别所在路径，数据格式需为SequenceFile(key, Cluster | Canopy)；**
- **-o：输出聚类后的结果存放路径**
- **-k：随机选取的类别数目**
- **-dm：采用的向量距离计算方法，在此指定实现类**
- **-x：最多迭代终止次数**
- **-cd：聚集偏移量，即本轮迭代次数与上一轮迭代相比，变动点总量变化值**
- **-ow：结果目录已经存在时是否对其覆盖**
- **-cl：是否首先通过canopy选取k个点，然后再运行k-means聚类**
- **-xm： sequential 或者 mapreduce，顺序执行（本地执行）还是分布式执行（采用mapreduce）**

Mahout自带的向量距离计算类

- 位于java包org.apache.mahout.distance中
  - ✓ 比如：org.apache.mahout.distance.CosineDistanceMeasure

Distance measure	Number of iterations	Vectors[a] in cluster 0	Vectors in cluster 1
EuclideanDistanceMeasure	3	0, 1, 2, 3, 4	5, 6, 7, 8
SquaredEuclideanDistanceMeasure	5	0, 1, 2, 3, 4	5, 6, 7, 8
ManhattanDistanceMeasure	3	0, 1, 2, 3, 4	5, 6, 7, 8
CosineDistanceMeasure	1	1	0, 2, 3, 4, 5, 6, 7, 8
TanimotoDistanceMeasure	3	0, 1, 2, 3, 4	5, 6, 7, 8

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2}$$

$$d = (a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2$$

$$d = |a_1 - b_1| + |a_2 - b_2| + \ldots + |a_n - b_n|$$

$$d = 1 - \frac{(a_1 b_1 + a_2 b_2 + \ldots + a_n b_n)}{\sqrt{(a_1^2 + a_2^2 + \ldots + a_n^2)} + \sqrt{(b_1^2 + b_2^2 + \ldots + b_n^2)} - (a_1 b_1 + a_2 b_2 + \ldots + a_n b_n)}$$

一个K-means聚类实例

步骤1：下载测试数据（格式为SGML）：

http://www.daviddlewis.com/resources/testcollections/reuters21578/reuters21578.tar.gz

步骤2：将数据解压

```
$ mkdir -p mahout-work/reuters-sgm
$ cd mahout-work/reuters-sgm && tar xzf ../reuters21578.tar.gz && cd .. && cd ..
```

步骤3：将SGML格式数据转化为文本文件

```
$ bin/mahout org.apache.lucene.benchmark.utils.ExtractReuters mahout-work/reuters-sgm mahout-work/reuters-out
```

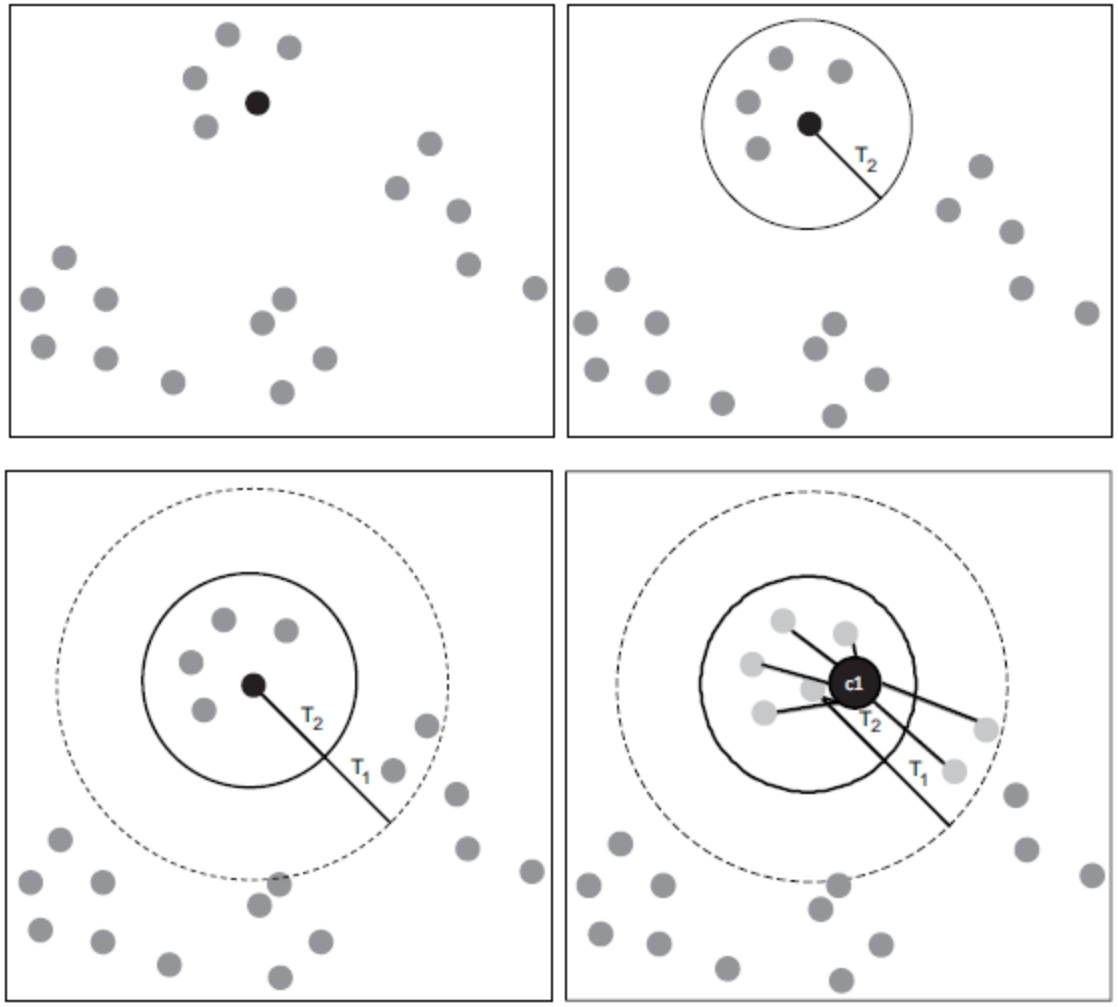
步骤4：将数据转化为SequenceFile格式

```
$ bin/mahout seqdirectory \
    -i mahout-work/reuters-out \
    -o mahout-work/reuters-out-seqdir \
    -c UTF-8 -chunk 5
```

步骤5：运行K-means

```
$ bin/mahout kmeans \
    -i mahout-work/reuters-out-seqdir-sparse-kmeans/tfidf-vectors/ \
    -c mahout-work/reuters-kmeans-clusters \
    -o mahout-work/reuters-kmeans \
    -dm org.apache.mahout.distance.CosineDistanceMeasure –cd 0.1 \
    -x 10 -k 20 –ow
```

寻找最优的初始点—Canopy算法



寻找最优的初始点—Canopy算法

```
$ bin/mahout kmeans \
    -i mahout-work/reuters-out-seqdir-sparse-kmeans/tfidf-vectors/ \
    -o reuters-kmeans-clusters \
    -dm org.apache.mahout.common.distance.TanimotoDistanceMeasure \
    -c reuters-canopy-centroids/clusters-0   \
    -cd 0.1 -ow -x 20 –cl
```

注：

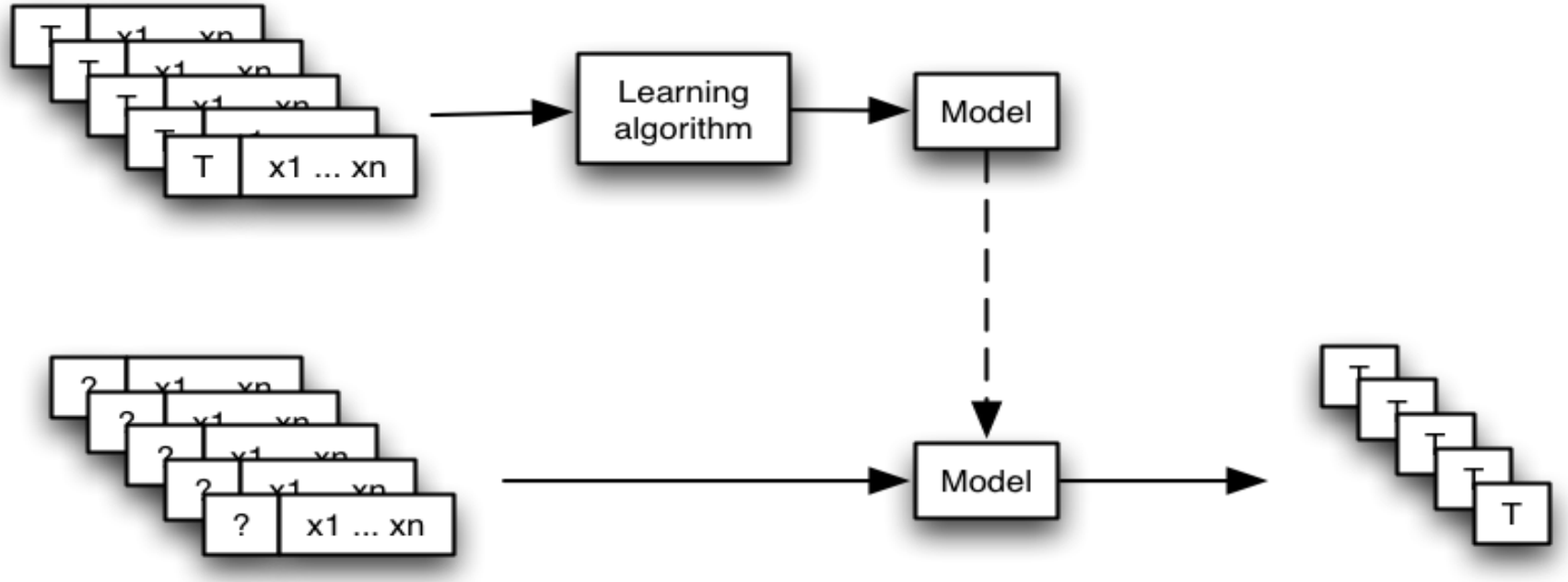
- 使用“-cl”参数，启用canopy算法
- 舍弃“-k”参数，否则canopy选取的点将被随机点覆盖

目录

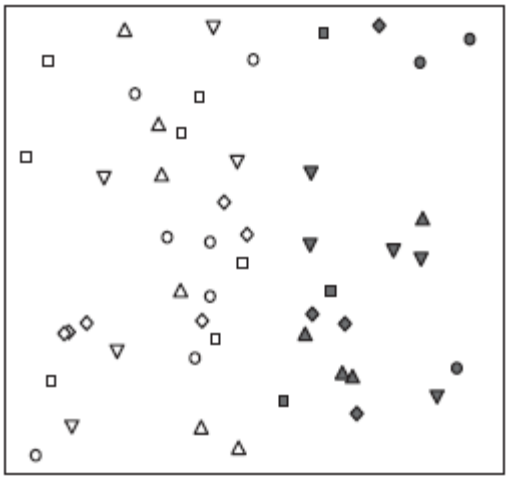
分类的基本流程

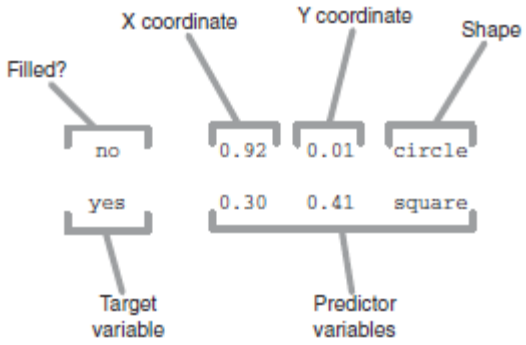
- 有监督机器学习算法
- 需提供样本，根据样本得到分类模型
- 分类三步骤
  - ✓ 步骤1：训练样本，得到分类模型；
  - ✓ 步骤2：对分类模型进行测试，并尝试调优
  - ✓ 步骤3：将分类模型用于线上产品中

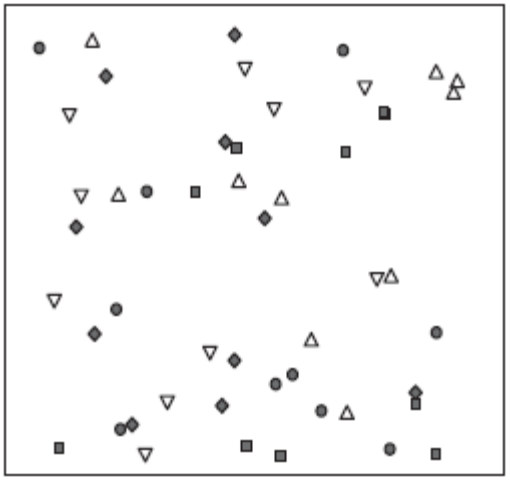
分类的基本流程

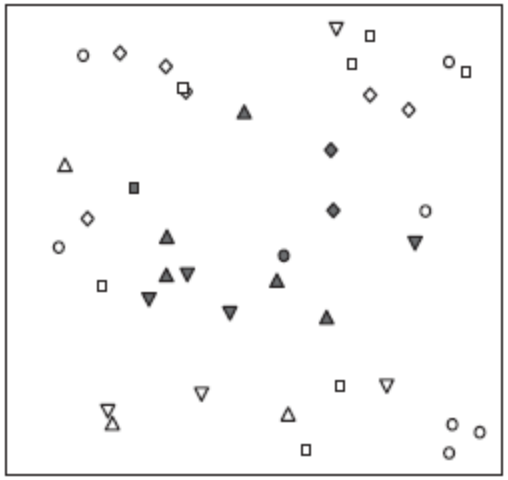


一个分类应用—判断坐标颜色









样本数据集准备

- 数据格式（**Mahout**压缩包的examples/src/main/resources/目录下）

```
"x","y","shape","color","k","k0","xx","xy","yy","a","b","c","bias"
0.923307513352484,0.0135197141207755,21,2,4,8,0.852496764213146,0.0124828536260896,0.000182782669907495,0.923406490600458,0.0778750292332978,0.644866125183976,1
0.711011884035543,0.909141522599384,22,2,3,9,0.505537899239772,0.64641042683833,0.826538308114327,1.15415605849213,0.953966686673604,0.46035073663368,1
0.75118898646906,0.836567111080512,23,2,3,9,0.564284893392414,0.62842000028592,0.699844531341594,1.12433510339845,0.872783737128441,0.419968245447719,1
0.308209649519995,0.418023289414123,24,1,5,1,0.094993188057238,0.128838811521522,0.174743470492603,0.519361780024138,0.808280495564412,0.208575453051705,1
0.849057961953804,0.500220163026825,25,1,5,2,0.720899422757147,0.424715912147755,0.250220211498583,0.985454024425153,0.52249756970547,0.349058031386046,1
0.0738831346388906,0.486534863477573,21,2,6,1,0.00545871758406844,0.0359467208248278,0.236716173379140,0.492112681164801,1.04613986717142,0.42632955896436,1
0.612888508243486,0.0204555552918464,22,2,4,10,0.375632323536926,0.0125369747681119,0.000418429742297785,0.613229772009826,0.387651566219268,0.492652707029903,1
0.207169560948387,0.932857288978994,23,2,1,4,0.0429192269835473,0.193259634985281,0.870222721601238,0.955584610897845,1.22425602987611,0.522604151014326,1
0.309267645236105,0.506309477845207,24,1,5,1,0.0956464763898851,0.156585139973909,0.256349287355886,0.593292308854389,0.856423069092351,0.190836685845410,1
0.78758287569508,0.171928803203627,25,2,4,10,0.620286786088131,0.135408181241926,0.0295595133710317,0.806130448165285,0.273277419610556,0.436273561610666,1
0.930236018029973,0.0790199618786573,21,2,4,8,0.86533904924026,0.0735072146828825,0.00624415437530446,0.93358620577618,0.105409523078414,0.601936228937031,1
0.238834470743313,0.623727766098455,22,1,5,1,0.0570419044152386,0.148967690904034,0.389036326202168,0.667890882268509,0.984077887735915,0.288991338582386,1
```

变量	解释	值范围
x	横坐标	浮点数，0-1
y	纵坐标	浮点数，0-1
shape	形状	形状编码，21-25
color	是否被填充	1—空的，2—被填充
k	利用x，y通过k-means聚类得到的类别ID	1-10
k0	利用x,y,color通过k-means聚类得到的类别ID	1-10
xx	X坐标的平方	浮点数，0-1
yy	X与y乘积	浮点数，0-1
yy	y坐标的平方	浮点数，0-1
a	与坐标（0,0）的距离	浮点数
b	与坐标（0,1）的距离	浮点数
c	与坐标（0.5,0.5）的距离	浮点数
bias	常量	1

步骤1：根据样本构建分类模型

```
$ bin/mahout trainlogistic --input donut.csv \
        --output ./model \
        --target color --categories 2 \
        --predictors x y --types numeric \
        --features 20 --passes 100 --rate 50
```

- **--input：指定样本数据集所在路径，为本地数据；**
- **--output：分类模型的存放目录**
- **--target：分类目标；**
- **-- categories ：分类目标包含的可能值数目**
- **-- predictors ：分类用到的变量值**
- **--types：变量值类型，可以是numeric, word或者text**

步骤2：对分类模型测试

**bin/mahout runlogistic --input donut.csv --model ./model **
--auc –confusion

输出：
AUC = 0.57
confusion: [[27.0, 13.0], [0.0, 0.0]]
...

- **--input：指定测试样本数据集所在路径，为本地数据；**
- **--model：分类模型的存放目录**
- **--auc：输出AUC值**
- **--confusion：打印混淆矩阵**

AUC和混淆矩阵的概念，可参考： http://bubblexc.com/y2011/148/

尝试其他分类模型

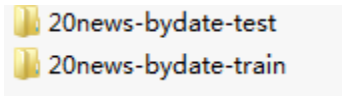
```
$ bin/mahout trainlogistic --input donut.csv --output model \
        --target color --categories 2 \
        --predictors x y a b c --types numeric \
        --features 20 --passes 100 --rate 50

$ bin/mahout trainlogistic --input donut.csv --output model \
        --target color --categories 2 \
        --predictors x y a b --types n --features 20 \
        --passes 100 --rate 50
```

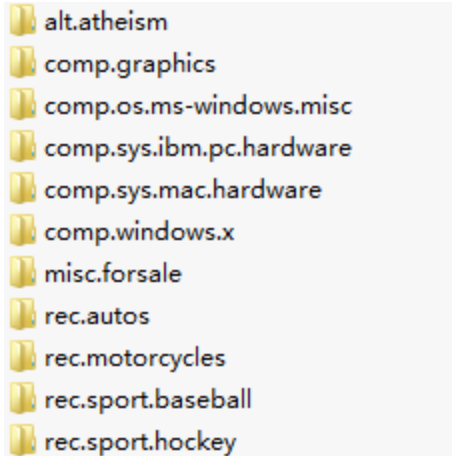
Mahout贝叶斯分类算法

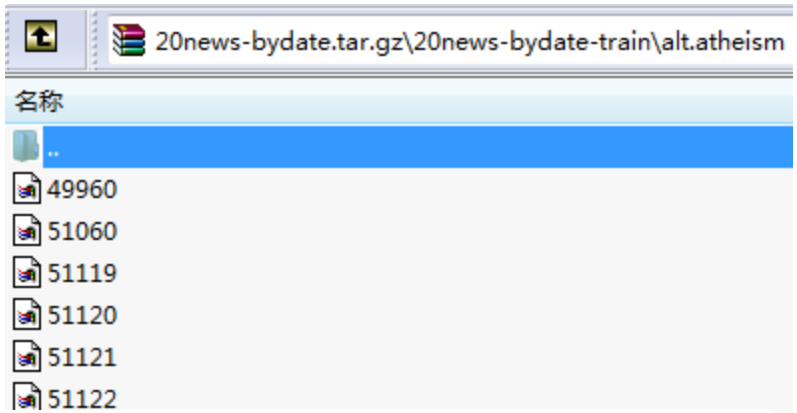
- **Naive Bayes（简称NB），朴素贝叶斯分类算法**
- **基本思想**
  - ✓ 已知类条件概率密度参数表达式和先验概率。
  - ✓ 利用贝叶斯公式转换成后验概率。
  - ✓ 根据后验概率大小进行决策分类。

数据集准备

http://people.csail.mit.edu/jrennie/20Newsgroups/20news-bydate.tar.gz







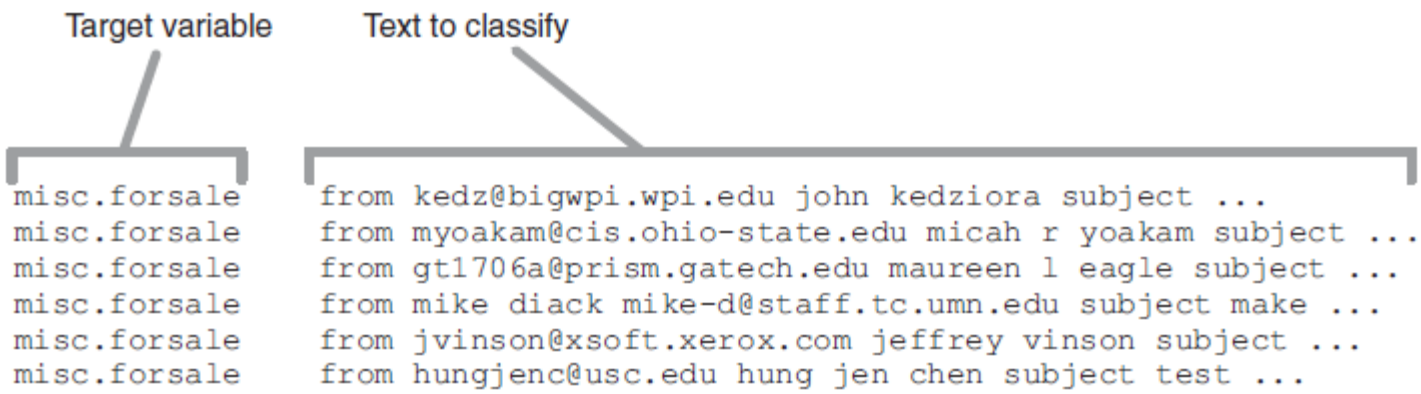

```
From: tedm@tsoft.net (Ted Matsumura)
Subject: Re: Windows gripe...
Organization: TSoft BBS and Public Access Unix, +1 415 969 8238
Lines: 37

In article <1ppmvf$qnh@bigboote.WPI.EDU> bigal@wpi.WPI.EDU (Nathan Charles Crowell) writes:
>
>Hi there,
>
>     There's one thing about Windows that really frosts me.
>I have 20MB of RAM installed in my system. I use a 5MB (2.5MB
>under Windows) disk-cache, and a 4MB permanent swap file.
>
>     While I can never fill the memory up, I still have problems
>sometimes because I run out of GDI resources. What gives?
>I think Windows could manage these resources a little better.
>
>     Does anyone have any input on how to conserve these resources
>so that I can avoid this problem?
```

数据预处理

```
$ bin/mahout prepare20newsgroups -p 20news-bydate-train/ \
        -o 20news-train/ \
        -a org.apache.lucene.analysis.standard.StandardAnalyzer \
        -c UTF-8
no HADOOP_CONF_DIR or HADOOP_HOME set, running locally
INFO: Program took 3713 ms
$ bin/mahout prepare20newsgroups -p 20news-bydate-test \
        -o 20news-test \
        -a org.apache.lucene.analysis.standard.StandardAnalyzer \
        -c UTF-8
```

- -p：训练或测试数据集位置；
- -o：输出数据存放位置
- -a：文本分析器
- -c：字符编码类型

数据预处理结果


Target variable
Text to classify
misc.forsale from kedz@bigwpi.wpi.edu john kedziora subject ...
misc.forsale from myoakam@cis.ohio-state.edu micah r yoakam subject ...
misc.forsale from gt1706a@prism.gatech.edu maureen l eagle subject ...
misc.forsale from mike diack mike-d@staff.tc.umn.edu subject make ...
misc.forsale from jvinson@xsoft.xerox.com jeffrey vinson subject ...
misc.forsale from hungjenc@usc.edu hung jen chen subject test ...

样本训练与模型测试

```
bin/mahout trainclassifier -i 20news-train \
        -o 20news-model \
        -type cbayes \
        -ng 1 \
        -source hdfs

bin/mahout testclassifier -d 20news-test \
        -m 20news-model \
        -type cbayes \
        -ng 1 \
        -source hdfs \
        -method sequential
```

目录

推荐算法介绍

- 诞生于电子商务系统中；
- 根据用户的兴趣特点和购买行为，向用户推荐用户感兴趣的信息和商品；
- 以协同过滤推荐算法为例进行介绍
  - ✓ 推荐系统中应用最早和最为成功的技术之一
  - ✓ 假设：为一用户找到他真正感兴趣的内容的好方法是首先找到与此用户有相似兴趣的其他用户，然后将他们感兴趣的内容推荐给此用 户

协同过滤推荐算法—应用案例

- **YouTube**
  - ✓ 使用者{观看、喜欢、评分}影片
- **Facebook**
  - ✓ 使用者{点击、点赞}连接或状态更新
- 新闻
  - ✓ 使用者{观看、喜欢、不喜欢}新闻

协同过滤推荐算法—基本元素

- Item
  - ✓能够被推荐给使用者的项目
- User
  - ✓能够推Item做评分，能为系统推荐Item的使用者
- Preference
  - ✓User对Item的评分
  - ✓{ userId, itemId, rating }

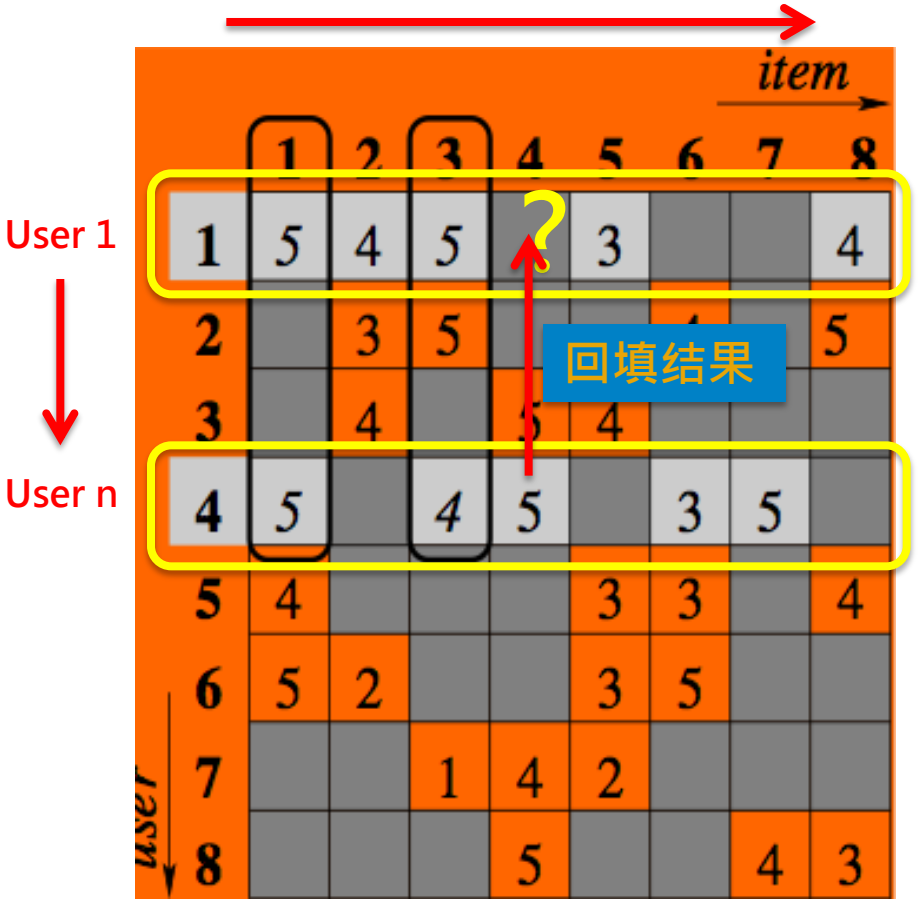
User-Item矩阵

	Item 1	Item 2	Item 3
User a	2	3	2
User b	4	?	3
User c	1	5	?

User-based

查找用户相似度

1. 如何预测用户1对于商品4的喜好程度?
2. 找到和用户1相似的用户且购买过商品4（基于购买记录）即为用户n
3. 根据用户n对商品4的评价，以相似度为权重回填结果
4. 针对所有用户组合，重复1~3，直到所有空格都被填满



Item-based

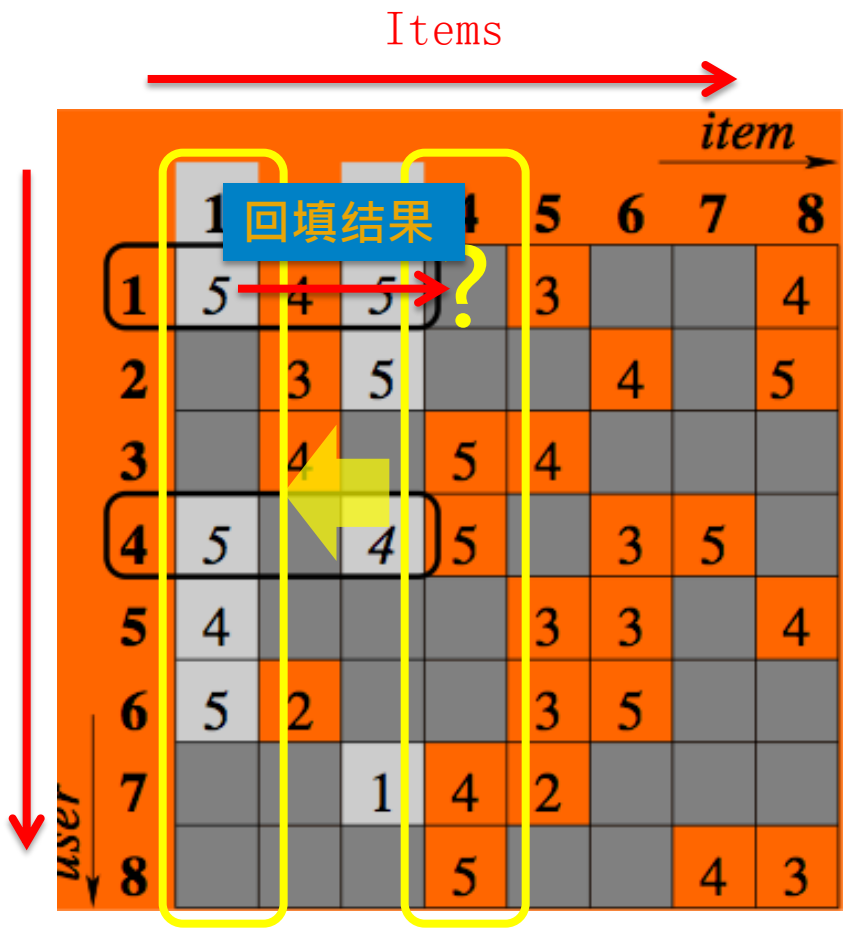
找到相似的Item

1.如何预测用户1对于商品4的喜好程度?

2.从用户1历史记录中，计算商品n和商品4的相似度（以其他用户的历史记录）

3.将用户1对于商品n的评价，以商品相似度为权重回填

4.针对所有商品组合，重复1~3直到所有空格都被填满



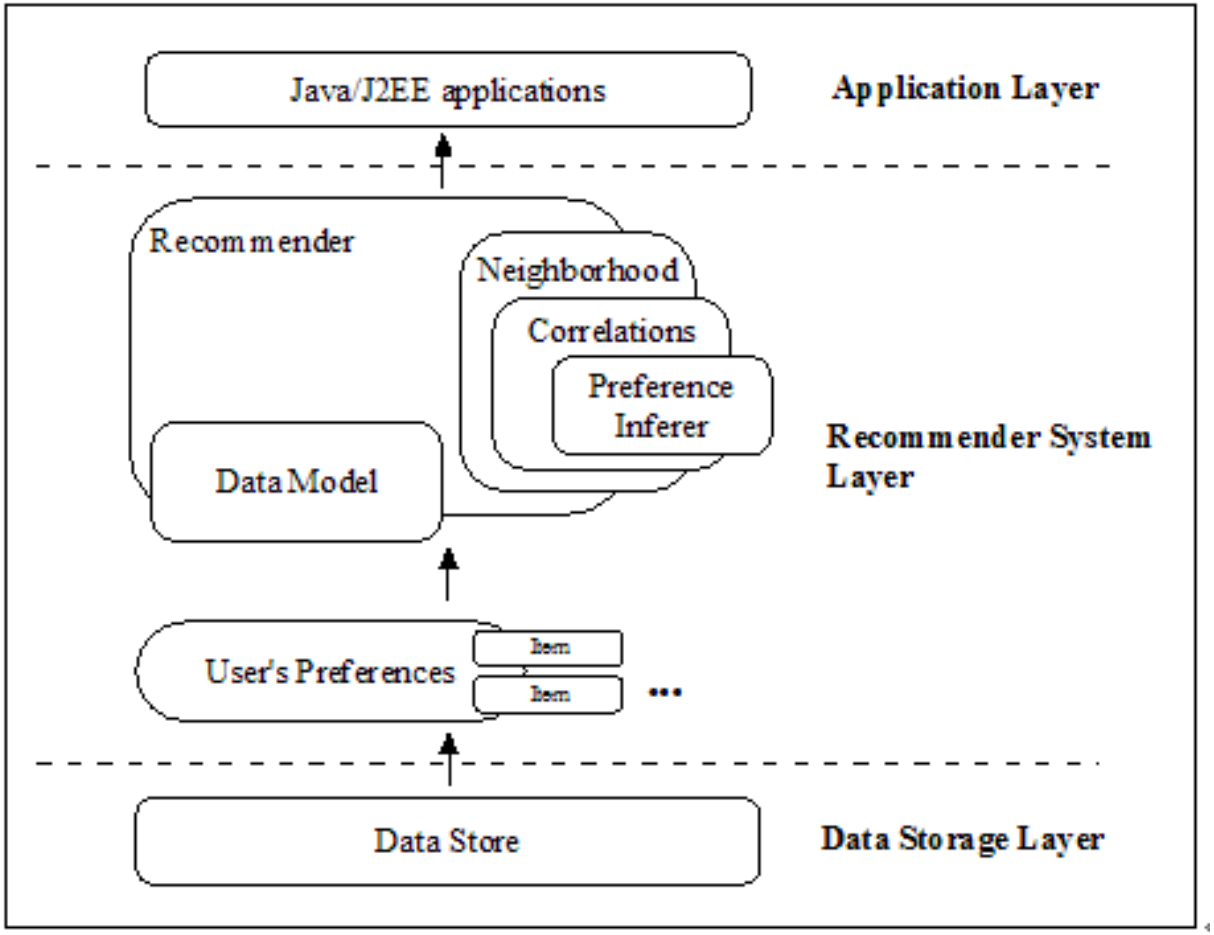
User-based 与 Item-based

- User-based
  - ✓基于使用者间的相似性推荐项目
- Item-based
  - ✓基于项目间的相似性推荐给使用者
- 各有优劣
  - ✓User-based的推荐效果好
  - ✓Item-based的计算效率高

Taste: Mahout自带的一个推荐系统实现

- Taste 是 Apache Mahout 提供的一个协同过滤算法的高效实现；
- 基于 Java 实现的可扩展的，高效的推荐引擎；
- 实现了最基本的基于用户的和基于内容的推荐算法，也提供了扩展接口，使用户可以方便的定义和实现自己的推荐算法；
- 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。

Taste: 基本架构



http://www.cnblogs.com/cajx/p/3279163.html

https://cwiki.apache.org/confluence/display/MAHOUT/Recommender+Documentation

数据集准备

- 三个不同规模数据集
  - ✓ http://grouplens.org/datasets/movielens/
- **MovieLens 1M**
  - ✓ 6,040个MovieLens用户针对3,900部电影；
  - ✓ 1,000,209条数据
- 包含三个文件
  - ✓ movies.dat ： 电影编号::电影名::电影类别
  - ✓ ratings.dat： 用户编号::电影编号::电影评分::时间戳
  - ✓ users.dat：用户编号::性别::年龄::职业::Zip-code

数据集准备

movies.dat

```
1::Toy Story (1995)::Animation|Children's|Comedy
2::Jumanji (1995)::Adventure|Children's|Fantasy
3::Grumpier Old Men (1995)::Comedy|Romance
4::Waiting to Exhale (1995)::Comedy|Drama
5::Father of the Bride Part II (1995)::Comedy
6::Heat (1995)::Action|Crime|Thriller
7::Sabrina (1995)::Comedy|Romance
8::Tom and Huck (1995)::Adventure|Children's
9::Sudden Death (1995)::Action
10::GoldenEye (1995)::Action|Adventure|Thriller
```

ratings.dat

```
1::1193::5::978300760
1::661::3::978302109
1::914::3::978301968
1::3408::4::978300275
1::2355::5::978824291
1::1197::3::978302268
1::1287::5::978302039
```

users.dat

```
1::F::1::10::48067
2::M::56::16::70072
3::M::25::15::55117
4::M::45::7::02460
5::M::25::20::55455
```

rating文件

- **Rating文件每一行数据格式为：**
  - **UserID::MovieID::Rating::Timestamp**
- **Mahout要求以下数据格式：**
  - **UserID,ItemID,Value**
- **可使用以下命令进行格式转化：**
  - **tr –s ‘:’ ‘,’ < ratings.dat | cut –f1-3 –d, > rating.csv**

rating文件

```
$ mahout recommenditembased \
        --input [input-hdfs-path] \
        --output [output-hdfs-path] \
        --usersFile [File listing users] \
        --numRecommendations [numRecommendations] \
        --similarityClassname  [similarityClassname] \
        --tempDir [tempDir]
```

- --input：输入数据存放路径；
- --output：输出数据存放位置
- --userFile：用户信息表
- --numRecommendations ：推荐产品数目
- -- similarityClassname：采用的相似度计算方法
- --tempDir：存放作业信息的临时目录

rating文件

```
mahout recommenditembased \
        --similarityClassname SIMILARITY_PEARSON_CORRELATION \
        --numRecommendations 2 \
        --usersFile input/users.csv \
        --input input/ratings.csv \
        --output output/
        --tempDir output/temp
```

- 计算结果格式如下：

UserID　　[ItemID:Weight, ItemID:Weight,…]

- 每行包含一个**userID**，后面是为该用户推荐的商品及权重

总结

1. 数据挖掘/机器学习算法对技术人员有较高要求；
2. Mahout提供了一个通用数据挖掘/机器学习库，但对技术人员要求仍非常高；