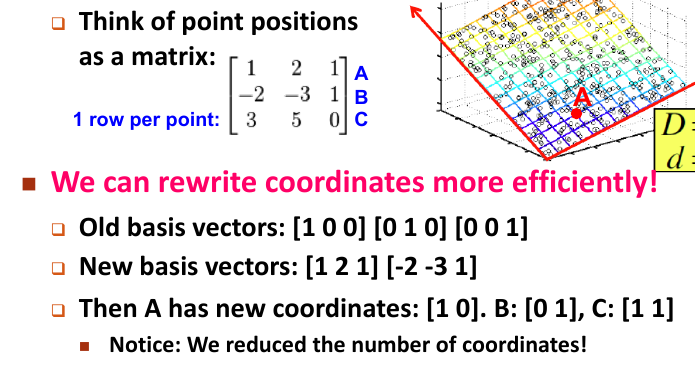
**知识点一：降维**

为什么要降维：数据子空间的轴可以有效地表示数据。

矩阵的秩（略） 作用。可以将矩阵的每一行（列）写成basis的线性组合形式。

Rank is dimensionality. 将三维空间的点的坐标用列向量表示，全部数据组合成一个矩阵。

上图三个点组成的3\*3矩阵，可以用旧的三维basis表示，由于其本身处于一个平面上，可以降维，只使用两个basis表示，构成新的坐标，每个点的新坐标，此时用新basis表示，只需要二维数据便可以表示。

数据降维：目的是寻找数据的新坐标轴。寻找数据间的隐藏关系（经常一次出现的单词）。去除冗余特征和噪声（并非所有数据都有用）。翻译和可视化。更方便的存储和处理。

接着上面的数据，将数据坐标降至一维。（降维会引发数据重叠，不精确问题）。

**降维方法1.SVD**

定义。

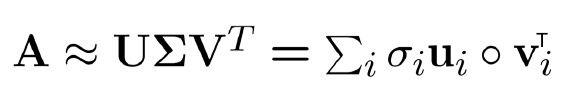


A要处理的数据矩阵。（m代表文本数量documents，n代表文本中的元素数量terms）。

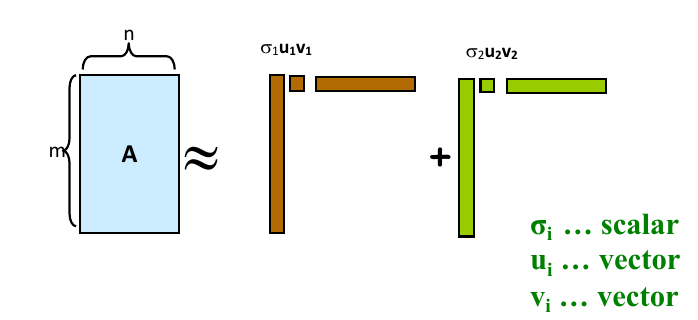
U 分解后的左奇异矩阵（m documents，r concepts）。

Σ 奇异值，从对角线延伸，是奇异值（strength of each concept）。

V 分解后的右奇异矩阵（r\*n的转置n\*r，n terms，r concepts）。



A可以表示为i个矩阵的和，每个矩阵是标量与uv向量的积



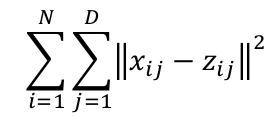
SVD性质

A总能找到SVD分解，左、右奇异矩阵是正交的，UTU=I，VTV=I，列之间是正交单位阵。

Σ奇异值为正，从上到下递减。

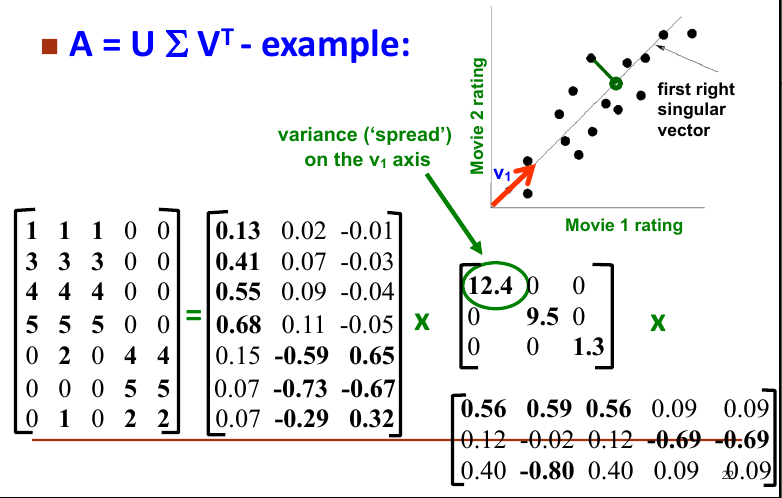
用SVD进行降维

将二维点坐标（x，y）降维到一维直线坐标。点的位置取决于它在方向向量v上的位置。如何选取方向向量——最小化reconstruction error



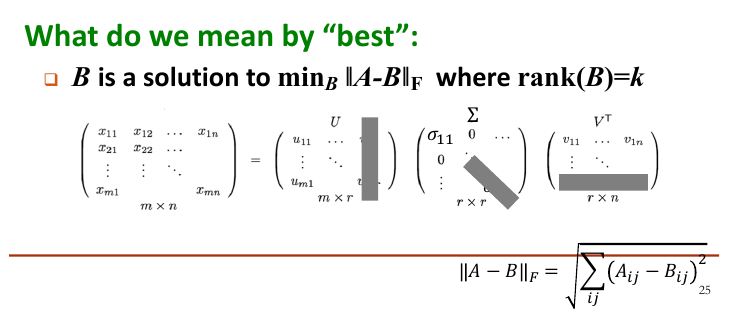
Xij是旧的二维坐标，Zij是新的坐标。

使用SVD得到的降维向量正好能将construction error最小化。取分解后的first right singular vector，右奇异矩阵的第一行向量。就是投影的方向。

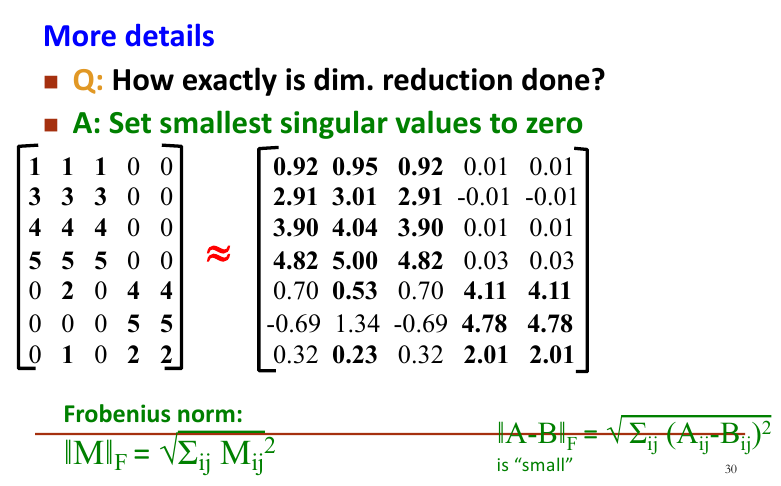


每个点代表着A矩阵的行向量。第一行等于U的第一、二、三列的第一个元素乘对应奇异值，再乘以V的第一二三列的每个元素得到的值相加。要是奇异值矩阵中，有比重很大的，可以只考虑它，其他的忽略。就变成在它上面的投影。

如何降维：



删除Σ的后面几个对角元素，可以降维。降维后A还是m\*n矩阵，但很容易能看出来，减去了很多不必要特征。

‘

为什么把小的奇异值变成0是正确的，因为ui和vi是单位向量，因此奇异值能够把它们成倍放大，把奇异值归零可以减小这些无关紧要的特征，zeroing small o introduces small error。

原则，应该保留多少个奇异值，keep 80%-90%energy，保留所有奇异值的平方和的百分之八十到九十。

SVD complexity

Min（O(nm2），O（n2m）)。当只需要求奇异值或前k个奇异向量或矩阵很稀疏时，less work。

提供功能的包：LINPACK，MATLAB，SPlus，Mathematica。

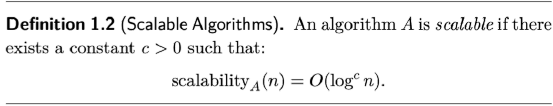
**分布式：MapReduce**

序列计算 在处理器上按照先后顺序进行 将问题分解为指令序列，按顺序依次执行指令，在一个处理器上执行所有指令，任意时间片里处理器只有一条指令在执行。

并行计算 将问题分解为可以并发处理的子问题 每一个子问题分为逻辑虚列 需要一个总体的控制机制

在大数据处理中 O(n2)算法很难处理intractable

算法可扩展（scalable）性



当c=0，A是线性scalable，当A（n）=o（1）是super scalable

大数据分析算法特点：以优化为中心，多轮迭代直到收敛，容错高。

参数更新方案：

同步更新Bulk synchronous parallel（BSP）

每一轮迭代设置同步点，统一更新参数，进入下一轮迭代。最大程度上模拟了序列计算，结果基本一致，具有同样的理论保障。

缺点 同步点的设置导致BSP低效，网络速度慢时延大，计算机速度不一致导致等待。

异步更新Async

完全没有参数同步，本地数据不全面，单个节点只有部分数据，独立更新参数，从整体上看结果不正确。不同机器上更新次数差异很大，参数相互依赖，参数过期。

半同步更新：综合上面两个。

MapReduce：如何解决局部服务器损坏问题。

问题：copy数据over network需要时间

解决：将电脑安置在靠近数据的地方 或 储存多份文件的拷贝

Storage infrastructure，file system

Google GFS。Hadoop：HDFS

Programming model

分布式文件系统

Chunk server 将文件分割成连续的chunks，一般每个chunk 16-64MB，每份chunk都复制两三份，让复制文件保存在不同的轨道上。

Master Node aka Name Node 储存metadata（元数据）的节点，可以被复制

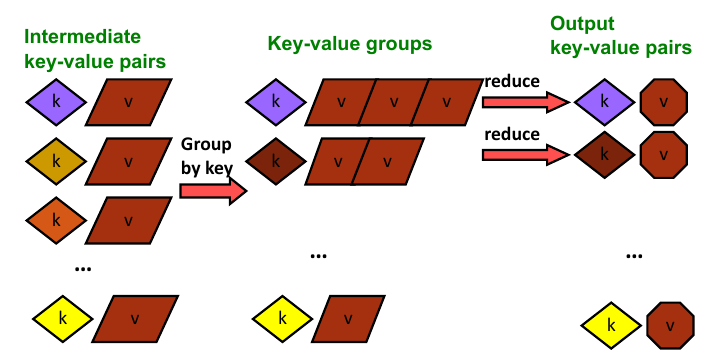
Client Library for file access 让master 寻找对应chunk server，直接和chunk server连接获取数据。

Map reduce过程 顺序读取数据。

Map 用户设计map function 提取感兴趣的信息

Group by keys sort and shuffle 系统检索所有键值，输出key-value

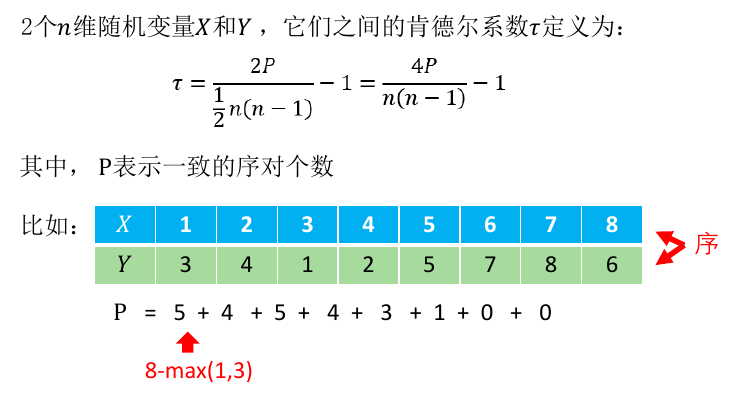
Reduce 设计reduce function： aggregate summarize filter transform



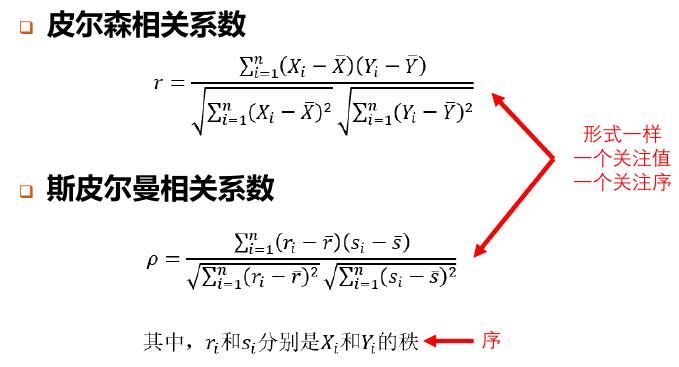
**相关性分析知识：**

传统统计相关性分析

肯德尔相关系数

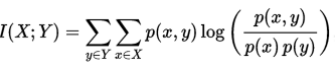


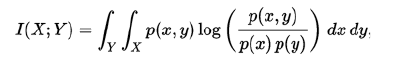
皮尔森相关系数 斯皮尔曼相关系数



大数据的统计相关性分析

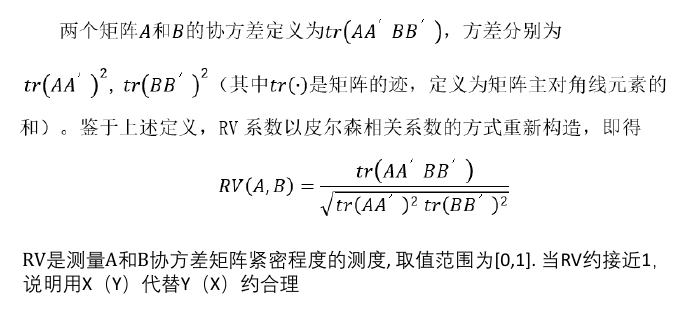
互信息 两个随机变量XY的互信息可以定义为

离散

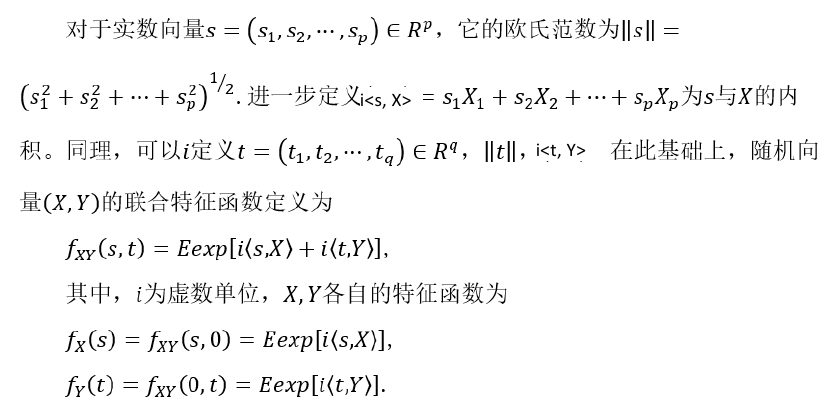
连续

基于矩阵计算的相关系数

RV系数



基于距离的相关系数



**Count Min sketch**

Sketch ： 是一种数据结构，压缩的，随机化的概括。

性质 逼近一个全面函数 holistic function 输入数据的次线性函数（sublinear function）输入数据的线性变换linear transform。

Count Min sketch（CMS）

没看懂 后面查

**Machine Learining**

Regression 统计方法，用来学习变量间的线性关系，在包含噪声的情况下

设置 一个对比的变量b 一系列预测用的变量a1---ad 假设b=a1x1----adxd+e

E是设计的noise 可以假设x0=0 使用b的n个观测值来逼近。

Least squares method

如何求解矩阵方程 那几种方法

Decision Tree

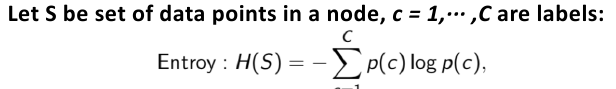
决策树 层层判断往下 得到最后的值

优点 非线性分类器 更好的可解释性 很自然地包含分类特征 category features

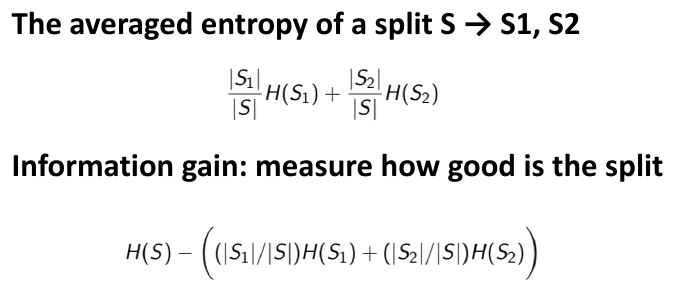
训练 训练速度慢 判断速度快 h次判断就能得到结果（h是树的深度，通常小于等于15）

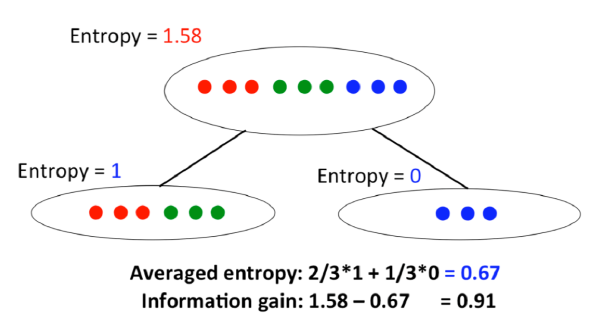
如何划分节点使得熵最小

熵



把S分割成S1，S2…时 每个S的信息量和得到的额外gain是





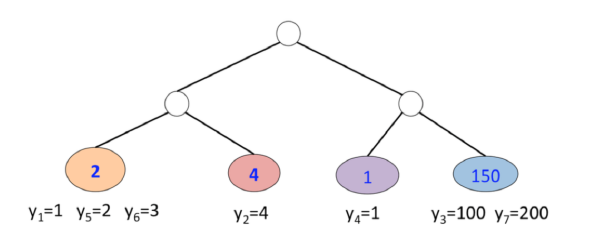
如何找到the best split 最好的分割

对于所有的特征和判断条件（threshold）：计算information gain after splitting，选择信息增益最大的一种划分。

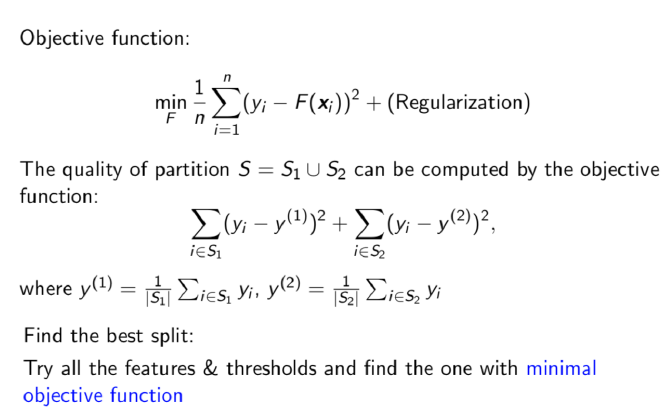
时间复杂度 对于n个样本的d种特征 O(nd)

回归树

在叶节点赋值为回归的那几个值的均值。



如何分割 目标函数



参数：最大深度 通常到10 每个节点的分支节点的最小数量（10，50,100）

Single decision tree 不是很好

我们可以使用多个决策树一起做决策。