数据分析及实践
Analysis and Practice of the Data

第二章 数据分析入门

刘 淇
Email: qiliuql@ustc.edu.cn

Data

- 数据采集　　Data Acquisition
- 数据预处理　　Data Preprocessing
- 特征工程　　Feature engineering

数据预处理：数据集成

- 数据集成：
  - 将多个数据源中的数据整合到一个一致的数据存储中
- 集成多个数据库时，经常会出现冗余数据
  - 相关分析冗余检测

$$r_{A,B} = \frac{\sum (A-\overline{A})(B-\overline{B})}{(n-1)\sigma_A \sigma_B}$$

冗余数据带来的问题：

浪费存储、重复计算

  - χ^2检验，值越大，两个变量相关的可能性越大

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})}{e_{ij}}$$

卡方检验：***oij*** 是联合事件 **(*Ai; Bj*)** 的观测频度（即实际计数），而 ***eij*** 是 **(*Ai; Bj*)**的期望频度。 卡方检验的原假设是 **A** 和 **B** 两个属性相互独立，如果可以拒绝该原假设，则我们说 **A** 和 **B** 是显著相关的。

数据预处理：数据集成

- 数据的距离度量（可以用来进行数据融合、去除冗余）
  - Euclidean Distance（欧几里得距离）

$$dist = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$$

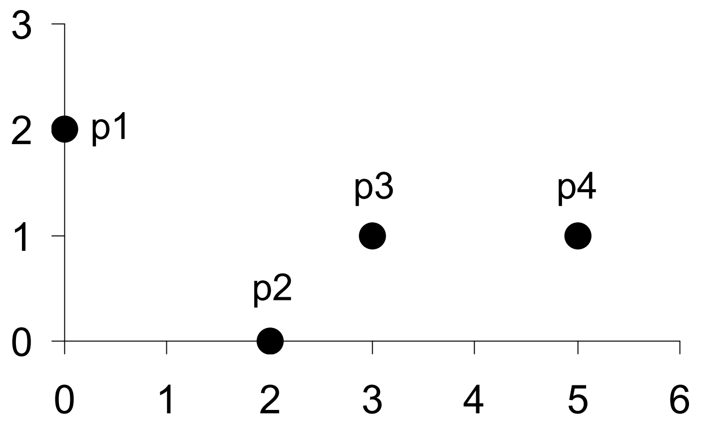
Where n is the number of dimensions (attributes) and p_k and q_k are, respectively, the kth attributes (components) or data objects p and q.

  - Standardization is necessary, if scales differ.
  -

数据预处理：数据集成

- 数据的距离度量
  - Euclidean Distance（欧几里得距离）

$$dist = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$$



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

数据预处理：数据集成

- 数据的距离度量
  - Minkowski Distance(明氏距离) is a generalization of Euclidean Distance

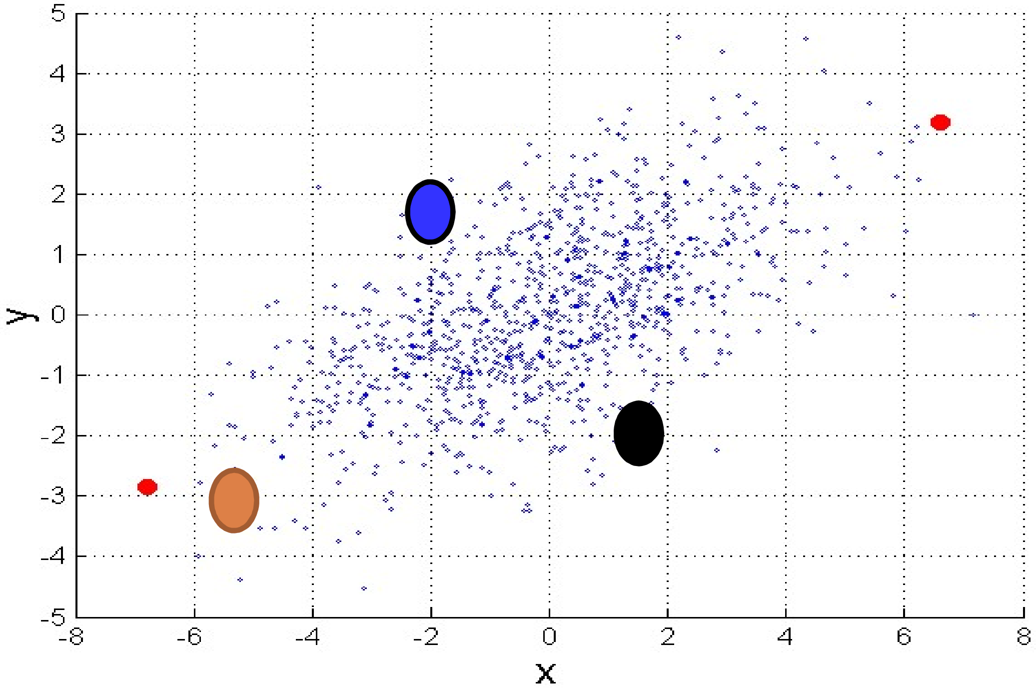
$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

Where r is a parameter, n is the number of dimensions (attributes) and pk and qk are, respectively, the kth attributes (components) or data objects p and q.

数据预处理：数据集成

- 数据的距离度量
  - 马氏距离

$$mahalanobis(p,q) = (p-q)\sum^{-1}(p-q)^T$$



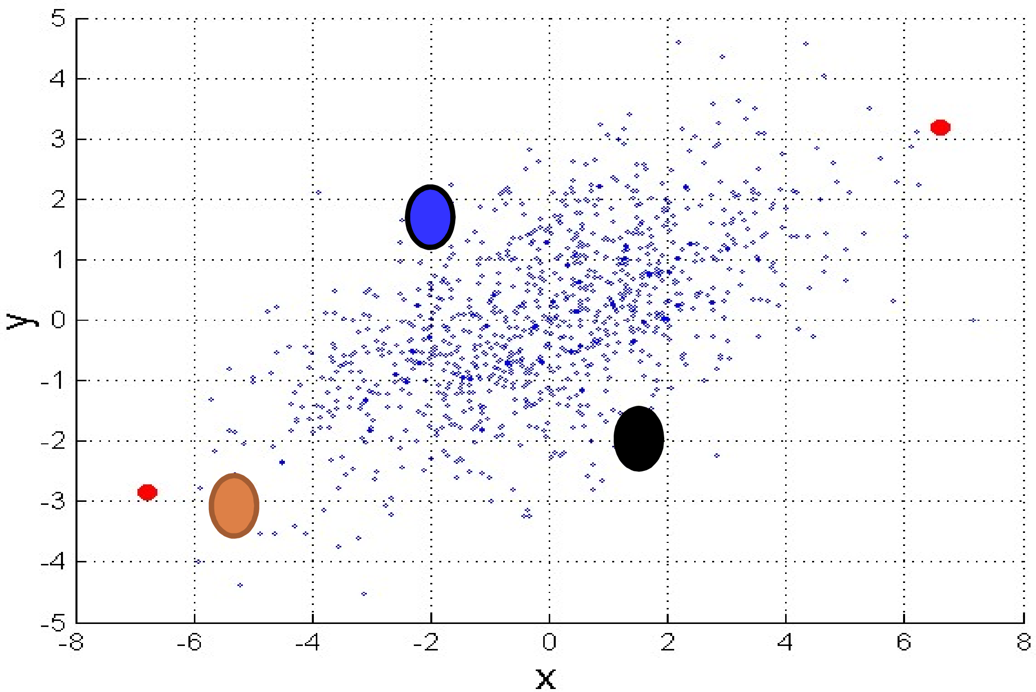
欧式距离有个默认的假设，就是属性之间是相互独立的，但是实际情况中，有些属性之间是不独立的，会相互关联（如身高体重），例如上图的**x**和**y**属性就是相互关联的，如果此时再用欧式距离，非独立属性的差异会被重复累计，是不合理的。我们可以乘以协方差的逆矩阵，相当于消除协方差对距离的影响，协方差越大（正数）说明这两个属性越相关，对应的逆就会越小，也就是距离的重复累计就越小。

数据预处理：数据集成

- 数据的距离度量
  - 马氏距离

表示数据的协方差距离，与欧氏距离不同的是它考虑到各种属性之间的联系（协方差，例如属性j的变化会带来属性k的变化）

$$mahalanobis(p,q) = (p-q)\Sigma^{-1}(p-q)^T$$



Σ 是总体样本 *X的协方差矩阵*

$$\Sigma_{j,k} = \frac{1}{n-1}\sum_{i=1}^{n}(X_{ij}-\overline{X}_j)(X_{ik}-\overline{X}_k)$$

Determining similarity of an unknown Sample set to a known one. It takes Into account the correlations of the Data set and is scale-invariant（独立于测量尺度）.

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.

数据预处理：数据集成

- 马氏距离，实例：

如果以厘米为单位来测量人的身高，以克（g）为单位测量人的体重。每个人被表示为一个两维向量，如一个人身高173cm，体重50000g，表示为（173,50000），根据身高体重的信息来判断体型的相似程度。

已知小明**（160,60000）**；小王**（160,59000）**；小李**（170，60000）**。根据常识可以知道小明和小王体型相似。但是如果根据欧氏距离来判断，小明和小王的距离要远大于小明和小李之间的距离，即小明和小李体型相似。这是因为不同特征的度量标准之间存在差异而导致判断出错。

以克（g）为单位测量人的体重，数据分布比较分散，即方差大，而以厘米为单位来测量人的身高，数据分布就相对集中，方差小。而且身高体重之间有一定的相关性。

马氏距离使得特征之间的关系更加符合实际情况。

数据预处理：数据集成

- 数据的距离度量
- Common situation is that objects, *p* and *q*, have only binary attributes（**0** 或 **1**）
- Compute similarities using the following quantities

  F01 = the number of attributes where *p* was 0 and *q* was 1

  F10 = the number of attributes where *p* was 1 and *q* was 0

  F00 = the number of attributes where *p* was 0 and *q* was 0

  F11 = the number of attributes where *p* was 1 and *q* was 1
- **Simple Matching** and **Jaccard** Coefficients（Jaccard系数）

  SMC = number of matches / number of attributes

  = (F11 + F00) / (F01 + F10 + F11 + F00)

  J = **number of 11 matches** / **number of non-zero attributes**

  = (F11) / (F01 + F10 + F11)

- 数据的距离度量

Simple Matching and **Jaccard** Coefficients（Jaccard系数）

p = 1 0 0 0 0 0 0 0 0 0
q = 0 0 0 0 0 0 1 0 0 1

F01 = 2 (the number of attributes where p was 0 and q was 1)
F10 = 1 (the number of attributes where p was 1 and q was 0)
F00 = 7 (the number of attributes where p was 0 and q was 0)
F11 = 0 (the number of attributes where p was 1 and q was 1)

SMC = (F11 + F00) / (F01 + F10 + F11 + F00)
= (0+7) / (2+1+0+7) = 0.7

$J = (F_{11}) / (F_{01} + F_{10} + F_{11}) = 0 / (2 + 1 + 0) = 0$

数据预处理：数据集成

- 数据的距离度量
  - Cosine Similarity（余弦相似性 ）
  - If d_1 and d_2 are two document vectors, then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1||\ ||d_2||,$$

where • indicates vector dot product and || d || is the length of vector d.

- Example:

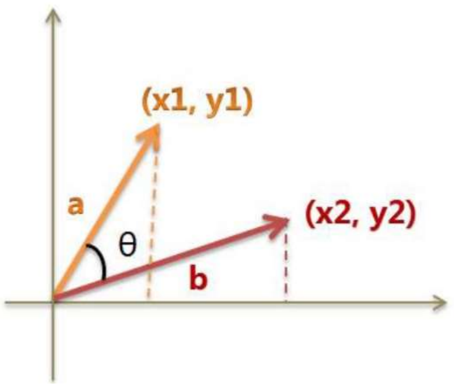
$$d_1 = \mathbf{3\ 2\ 0\ 5\ 0\ 0\ 0\ 2\ 0\ 0}$$
$$d_2 = \mathbf{1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 2}$$

$d_1 \bullet d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$||d_1||$ = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)$^{0.5}$ = (42) $^{0.5}$ = 6.481

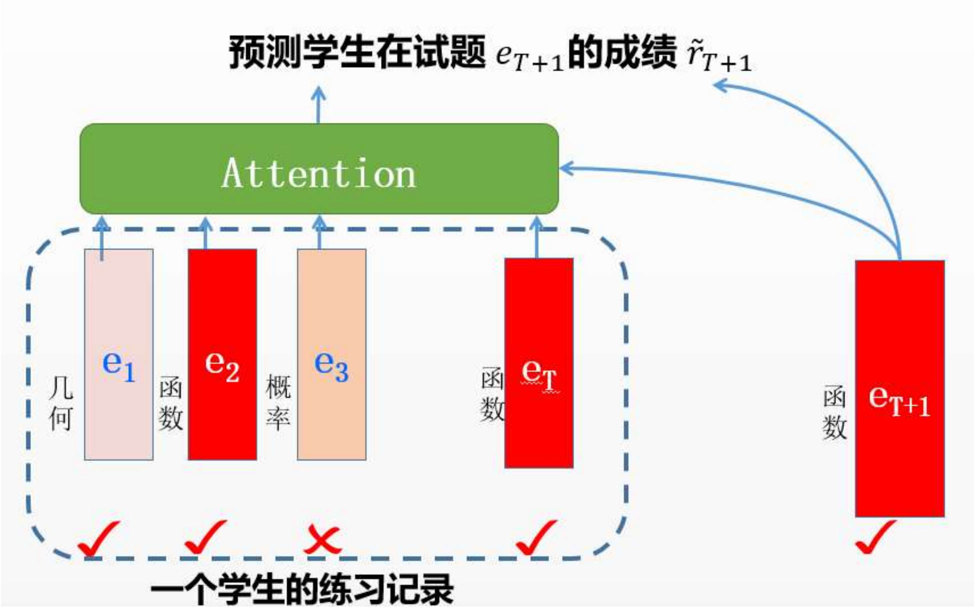
$||d_2||$ = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2) $^{0.5}$ = (6) $^{0.5}$ = 2.245

$\cos(d_1, d_2)$ = .3150



数据预处理：数据集成

- 数据的距离度量
  - **Cosine Similarity（余弦相似性 ）**
    - 推荐系统中，判断用户兴趣向量(User)与产品向量(Item)的相似度
    - 深度学习中，训练Attention（注意力机制）的权重
      - 基于注意力机制的学生成绩预测模型



Yu Su, Qi Liu, Zhenya Huang, Enhong Chen, et al. Exercise-Enhanced Sequential Modeling for Student Performance Prediction, AAAI 2018.

数据预处理：数据集成

- 数据的距离度量
  - **Correlation(相关度)** measures the linear relationship between objects
  - To compute correlation, we standardize data objects, p and q, and then take their dot product
  - 可以简单理解为：p和q的协方差/(p的标准差*q的标准差)

$$p'_k = (p_k - mean(p)) / std(p)$$

$$q'_k = (q_k - mean(q)) / std(q)$$

$$correlation(p, q) = p' \bullet q' / (n - 1)$$

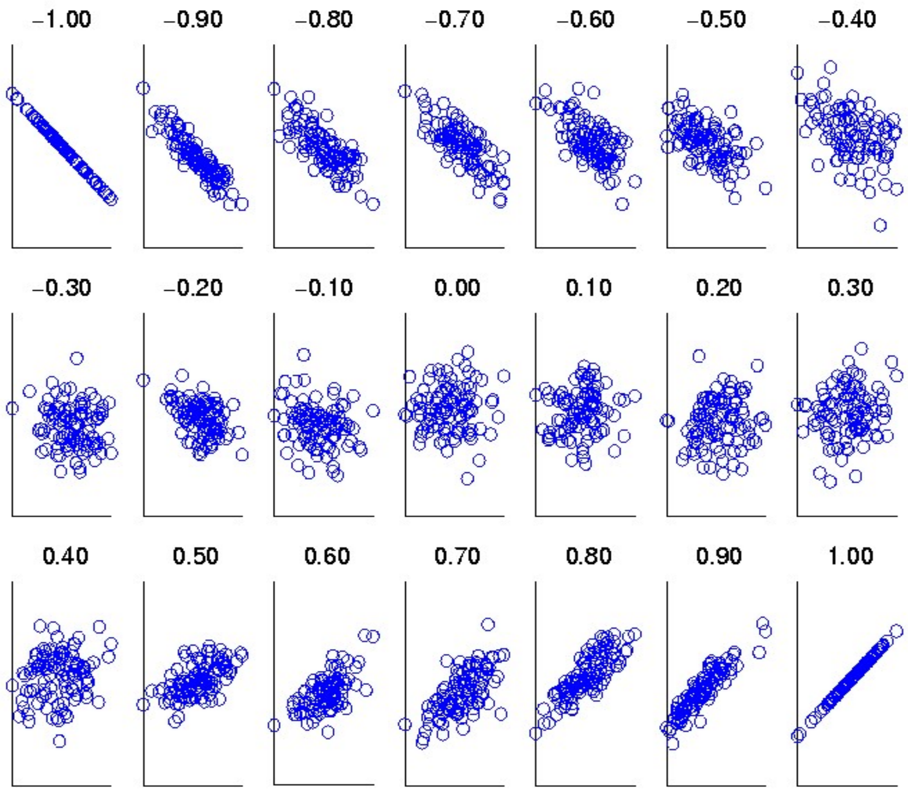
数据预处理：数据集成

- 数据的距离度量
  - **Correlation** measures the linear relationship between objects
  - To compute correlation, we standardize data objects（z-score），p and q, and then take their dot product
  - 可以简单理解为：p和q的协方差/(p的标准差*q的标准差)

$$corr(p,q) = \frac{\sum_k (p_k - \bar{p})(q_k - \bar{q})}{\sqrt{\sum (p_k - \bar{p})^2}\sqrt{\sum (q_k - \bar{q})^2}}$$

数据预处理：数据集成

- 数据的距离度量 **Correlation measures the linear relationship between objects**



Scatter plots showing the similarity from –1 to 1.

两个数据对象x,y各有30个属性，这些属性值随机产生，使得x和y的相关度从-1到1，图中每个小圆圈代表30个属性中的一个，其x坐标是x的一个属性的值，而y坐标是y的相同属性的值

数据预处理：数据集成

- 数据的距离度量

  Correlation measures the linear relationship between objects

- X = (-3, -2, -1, 0, 1, 2, 3)
- Y = (9, 4, 1, 0, 1, 4, 9)

X与Y有没有关系?

$$corr(p,q) = \frac{\sum_k (p_k - \bar{p})(q_k - \bar{q})}{\sqrt{\sum (p_k - \bar{p})^2} \sqrt{\sum (q_k - \bar{q})^2}}$$

- Mean(X) = 0, Mean(Y) = 4
- Correlation= ?
  - = (-3)(5)+(-2)(0)+(-1)(-3)+(0)(-4)+(1)(-3)+(2)(0)+3(5)= 0

数据预处理：数据集成

- 数据的距离度量
  - May not want to treat all attributes the same.
    - Use weights w_k which are between 0 and 1 and sum to 1.

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^{n} \delta_k}$$

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} w_k |x_k - y_k|^r \right)^{1/r}$$

数据预处理：数据集成

- 数据的距离度量---练习题1
- 对于下面的x和y，计算指定的相似性或距离度量。余弦相似度、相关度、欧几里得距离、Jaccard。
  - X和Y是什么相关关系？

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / \|d_1\| \|d_2\|$$

X = (0, 1, 0, 1), Y = (1, 0, 1, 0)

$$corr(p,q) = \frac{\sum_k (p_k - \bar{p})(q_k - \bar{q})}{\sqrt{\sum (p_k - \bar{p})^2}\sqrt{\sum (q_k - \bar{q})^2}}$$

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

J = (F11) / (F01 + F10 + F11)

数据预处理：数据集成

数据的距离度量

- 无序数据：每个数据样本的不同维度是没有顺序关系的
  - 余弦相似度、相关度、欧几里得距离、Jaccard
- 有序数据：对应的不同维度(如特征)是有顺序(rank)要求的
  - 在信息检索中，如何判断不同检索方法返回的页面序列的优劣
  - 在推荐系统中，如何判断不同推荐序列的好坏
    - **Spearman Rank(斯皮尔曼等级)相关系数**
    - 归一化的折损累计增益(NDCG)
    - 肯德尔相关性系数
      - kendall correlation coefficient

i	相关度
1	3
2	3
3	2
4	0
5	1
6	2

方法返回结果

i	相关度
1	3
2	3
3	2
4	2
5	1
6	0

真实结果

数据预处理：数据集成

数据的距离度量—举例

- 已知6个候选网页与给定网页的相关度是3，2，3，0，1，2，所以在信息检索中，最好的返回结果应当如(a)所示。如果我们设计了两个检索算法，它们的返回结果分别是(b)和(c)，请问哪个方法的结果与真实结果更相似？

i	相关度
1	3
2	3
3	2
4	2
5	1
6	0

(a)真实结果

i	相关度
1	3
2	3
3	0
4	2
5	2
6	1

(b)方法1返回结果

i	相关度
1	3
2	3
3	2
4	0
5	2
6	1

(c)方法2返回结果

数据预处理：数据集成

- 有序数据的距离度量(信息检索、推荐系统等)
  - **Spearman Rank(斯皮尔曼等级)相关系数**
    - 比较两组变量的相关程度
    - 当关系是非线性时，它是两个变量之间关系评价的更好指标

$$\rho_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

    - ρ_s：表示斯皮尔曼相关系数
    - d_i^2：表示每一对样本之间等级的差
    - n：表示样本容量
  - ρ_s的范围：-1 to 1 (正相关:$\rho_s > 0$，负相关:$\rho_s < 0$，不相关:$\rho_s = 0$)

数据预处理：数据集成

- **有序数据的距离度量(信息检索、推荐系统等)**
  - **Spearman Rank(斯皮尔曼等级)相关系数**
  - X = (a, b, c, d, e, f)
  - Y = (c, a, e, d, f, b)

$$\boldsymbol{d_i = Y_i - X_i}$$

  - $d_i^2 = (4, 1, 4, 0, 1, 16)$

$$\rho_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

  - $\rho = 1 - \frac{6(26)}{6(36-1)} \approx 1 - 0.743 = 0.257$

数据预处理：数据集成

数据的距离度量—课后思考

- Spearman Rank相关度与Pearson相关度之间的联系与区别？

$$\rho_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

$$corr(p,q) = \frac{\sum_k (p_k - \bar{p})(q_k - \bar{q})}{\sqrt{\sum (p_k - \bar{p})^2}\sqrt{\sum (q_k - \bar{q})^2}}$$

斯皮尔曼相关系数被定义成等级数据变量
(rank/order variables)之间的皮尔逊相关系数

数据预处理：数据集成

$$\rho_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

数据的距离度量--练习题2

- 已知6个网页与给定网页的相关度是3，2，3，0，1，2，所以在信息检索中，最好的返回结果应当如(a)所示。如果我们设计了两个检索算法，它们的返回结果分别是(b)和(c)，请问哪个方法的结果与真实结果更相似（给出**Spearman**计算结果）。

i	相关度
1	3
2	3
3	2
4	2
5	1
6	0

(a)真实结果

i	相关度
1	3
2	3
3	0
4	2
5	2
6	1

(b)方法**1**返回结果

i	相关度
1	3
2	3
3	2
4	0
5	2
6	1

(c)方法**2**返回结果

只考虑了每个位置**(entry)**的数据与真实数据的顺序差异，但是没有考虑到不同位置**(entry)**的重要性差异

数据预处理：数据集成

- **有序数据的距离度量(信息检索、推荐系统等)**
  - **NDCG(Normalized Discounted cumulative gain)**
    - **CG(累计增益)**：只考虑到了相关性的关联程度，没有考虑每个推荐结果处于不同位置对整个推荐效果的影响

$$\mathrm{CG_k} = \sum_{i=1}^{k} \mathrm{rel_i}$$

rel_i表示处于位置 ii 的推荐结果的相关性

    - **DCG(折损累计增益)**：就是在每一个CG的结果上处以一个折损值，目的就是为了让排名越靠前的结果越能影响最后的结果

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

    - i表示推荐结果的位置，i越大，则推荐结果在推荐列表中排名越靠后推荐效果越差，DCG越小

数据预处理：数据集成

- 有序数据的距离度量(信息检索、推荐系统等)
  - **NDCG(** Normalized Discounted cumulative gain **)**
    - **NDCG**：由于搜索结果随着检索词的不同，返回的数量不一致，而DCG是一个累加的值，没法针对两个不同的搜索结果进行比较，因此需要归一化处理，这里是除以IDCG:

$$NDCG_k = \frac{DCG_k}{IDCG_k}$$

IDCG为理想（ideal）情况下最大的DCG值，指推荐系统为某一用户返回的最好推荐结果列表(或者，真实的数据序列)

数据预处理：数据集成

- 例，假设搜索返回的6个物品，其相关性分别是 3、2、3、0、1、2
  - CG@6 = 3+2+3+0+1+2
  - DCG@6 = 7+1. 89+3.5+0+0. 39+1.07 = 13.85
- 假如我们实际召回了8个物品，除了上面的6个，还有两个物品，第7个相关性为3，第8个相关性为0。那么在理想情况下的相关性分数排序应该是：3、3、3、2、2、1、0、0。计算IDCG@6：
  - IDCG = 7+4.42+3.5+1.29+1.16+0.36 = 17.73
- 可以计算：
- NDCG@6 = 13.85/17.73 = 0.78

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)}$$

i	rel
1	3
2	2
3	3
4	0
5	1
6	2

方法返回结果

i	rel
1	3
2	3
3	3
4	2
5	2
6	1

真实结果

数据预处理：数据集成

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)}$$

数据的距离度量--练习题3

- 已知6个网页与给定网页的相关度分别是3，2，3，0，1，2，所以在信息检索中，最好的返回结果应当如(a)所示。如果我们设计了两个检索算法，它们的返回结果分别是(b)和(c)，请问哪个方法的结果与真实结果更相似（根据**NDCG**的计算结果）。

i	相关度
1	3
2	3
3	2
4	2
5	1
6	0

(a)真实结果

i	相关度
1	3
2	3
3	0
4	2
5	2
6	1

(b)方法**1**返回结果

i	相关度
1	3
2	3
3	2
4	0
5	2
6	1

(c)方法**2**返回结果

可以只列出计算公式，不用给出计算结果

数据预处理：数据变换

- 数据变换的目的是将数据转换或统一成适合挖掘的形式。
  - 规范化：将数据按比例缩放，使之落入一个小的特定区间
    - 最小－最大规范化
    - z-score规范化
    - 小数定标规范化
  - 离散化
    - 非监督离散化
    - 监督离散化
    - 相关性度量离散化

数据预处理：数据变换-规范化

- 最小一最大规范化

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score规范化（均值、标准差）
  - 最大最小值未知，或者离群点影响较大的时候适用

$$v' = \frac{v - \overline{A}}{\sigma_A}$$

- 小数定标规范化

$$v' = \frac{v}{10^j}$$

其中，j是使 Max(| v'|)<1的最小整数

数据预处理：数据变换-规范化

- 最小一最大规范化
  - 例：假设某属性规格化前的取值区间为[－100，100]，规格化后的取值区间为[0，1]，采用最小-最大规格化66，得

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

$$v' = \frac{66 - (-100)}{100 - (-100)}(1 - 0) + 0 = 0.83$$

采用最小-最大规格化**-80**？

数据预处理：数据变换-规范化

- z-score规范化
  - 最大最小值未知，或者离群点影响较大的时候适用

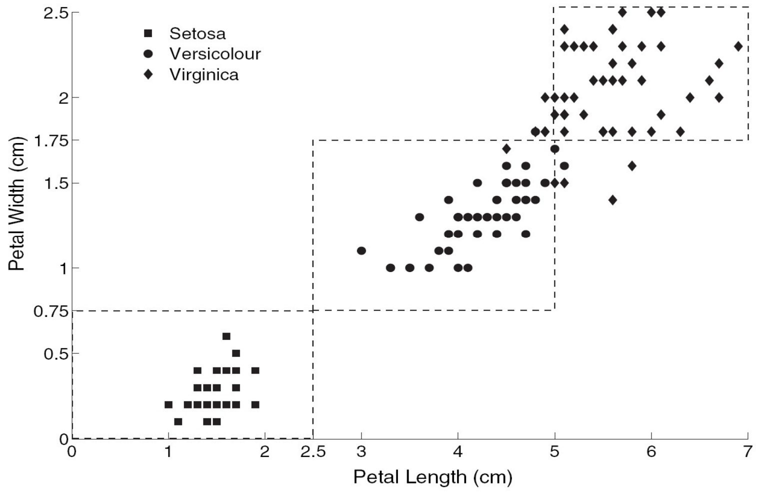
$$v' = \frac{v - \overline{\mathrm{A}}}{\sigma_A}$$

  - 例：假设某属性的平均值、标准差分别为80、25，采用零-均值规格化66

$$v' = \frac{66 - 80}{25} = -0.56$$

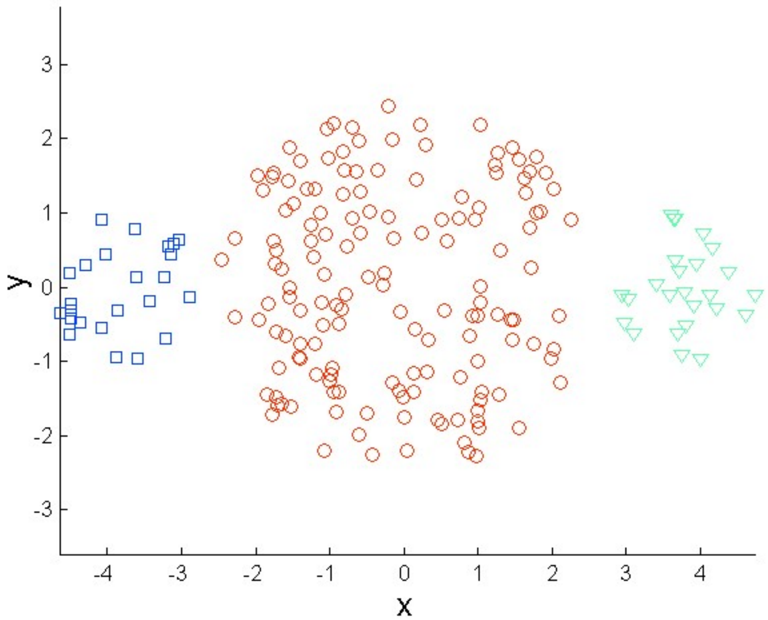
数据预处理：数据变换-离散化

- 非监督离散化
  - 分箱
  - 聚类
- 监督离散化
  - 基于熵的离散化，递归划分区间i，使得每一次划分点的熵最小：$e_i = -\sum_{j=1}^{k} p_{ij} log_2 p_{ij}$
- 相关性度量离散化
  - 基于χ^2分析的区间合并方法



数据预处理：数据变换-离散化

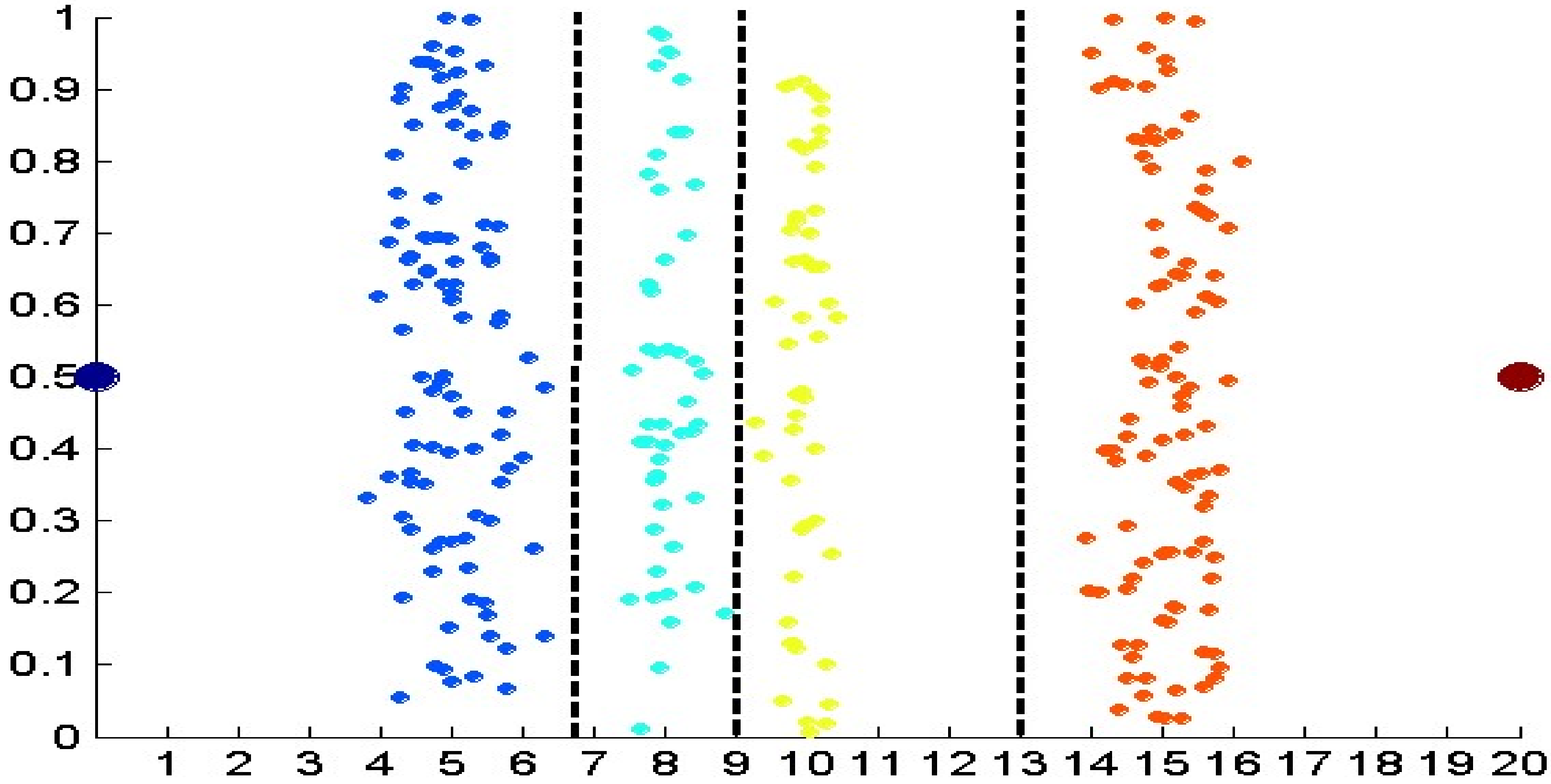
- 有监督的离散化
  - 基于熵的离散化



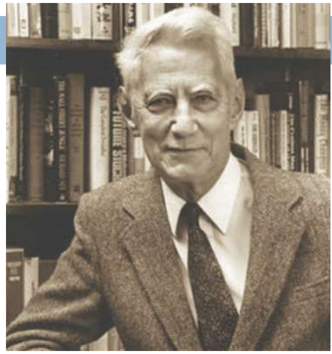
Original Points

数据预处理：数据变换-离散化

- 有监督的离散化
  - 基于熵的离散化



数据预处理：数据变换-离散化



- 熵—计算不纯性
  - Entropy at a given box（区间） t:

$$\boxed{Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)}$$

- (NOTE: p(j | t) is the relative frequency of class j at box t).

一般对数log是以2为底的

- Maximum (log n_c) when records are equally distributed among all classes implying least information 区间里面不同类别的样本均匀分布时，熵值最大（最不纯），为样本类数**n_c**的**log**

- Minimum (0.0) when all records belong to one class, implying most information 区间里面只有一类样本时，熵值最小（最纯）

数据预处理：数据变换-离散化

- 计算单个区间的 Entropy

$$Entropy(t) = -\sum_j p(j\,|\,t)\log_2 p(j\,|\,t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0　　P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6　　P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6　　P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

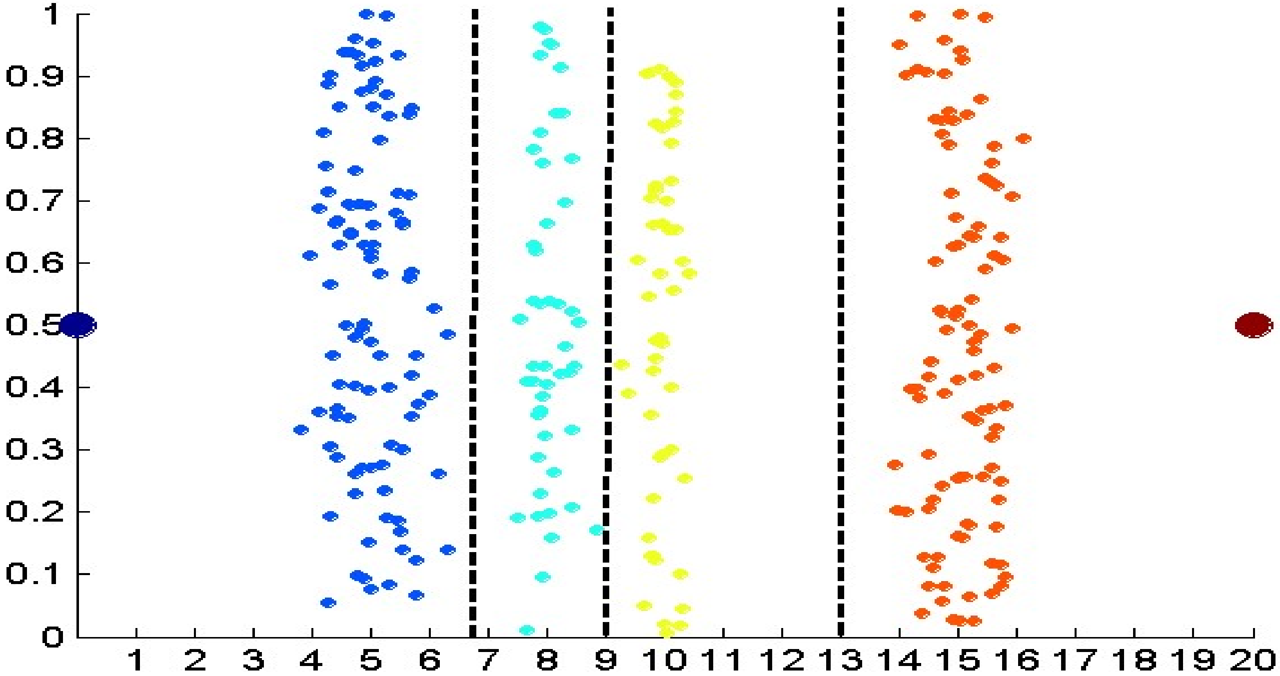
练习4:(1)如果区里t里面C1和C2的样本数各为3，Entropy是多少?
（2）如果区间t里面有四个类，而且样本数一样，Entropy是多少?

数据预处理：数据变换-离散化

- 根据Enttropy进行二分离散化

$$Entropy\ (t) = -\sum_j p(j\,|\,t)\log_2 p(j\,|\,t)$$

  - 先找到一个分隔点（属性值），把所有数据分到两个区间
  - 分别对两个子区间的数据进行二分隔
  - 重复以上步骤

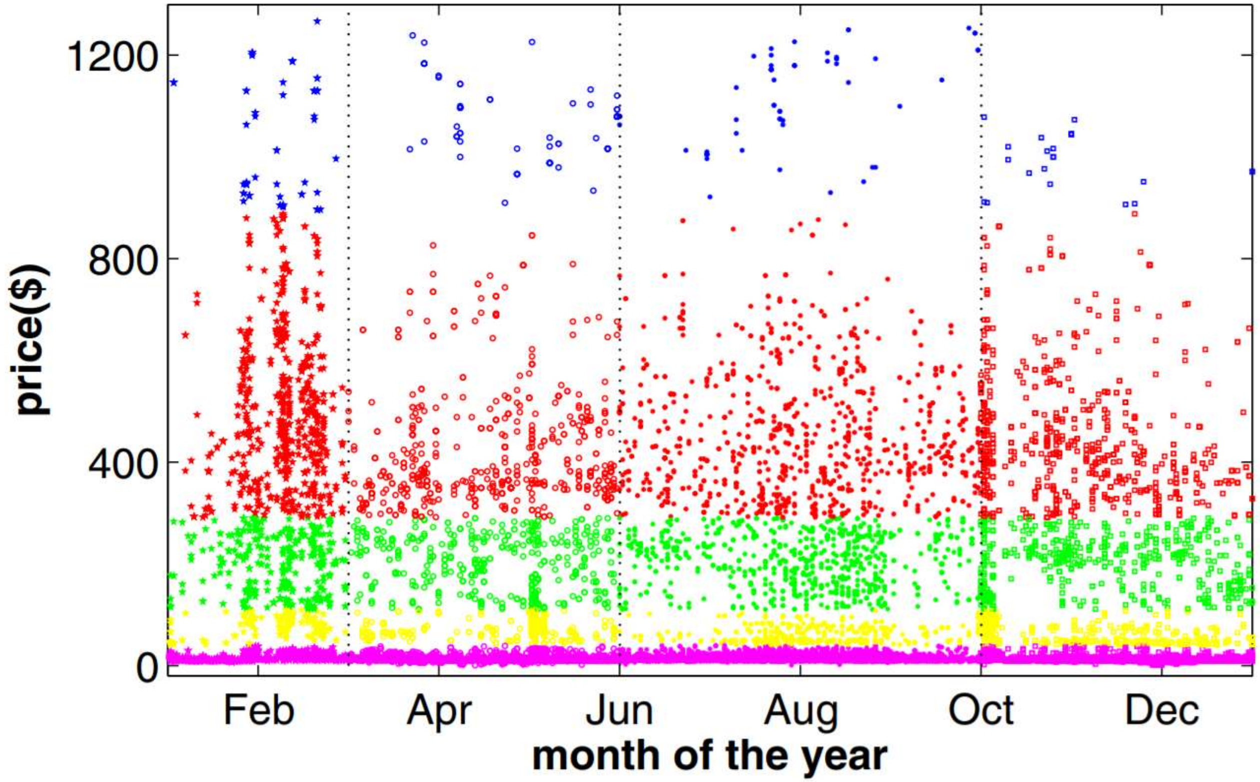


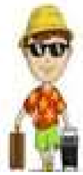
数据预处理：数据变换-离散化

- 熵（Entropy）的应用举例

$$WAE(i;S^P)=\frac{|S_1^P(i)|}{|S^P|}Ent(S_1^P(i))+\frac{|S_2^P(i)|}{|S^P|}Ent(S_2^P(i))$$

  - 使用熵进行旅游季节（Travel Season）的划分





Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. Personalized Travel Package Recommendation. **ICDM'2011**:407-416,(**Best Research Paper Award**)

数据预处理：数据变换-离散化

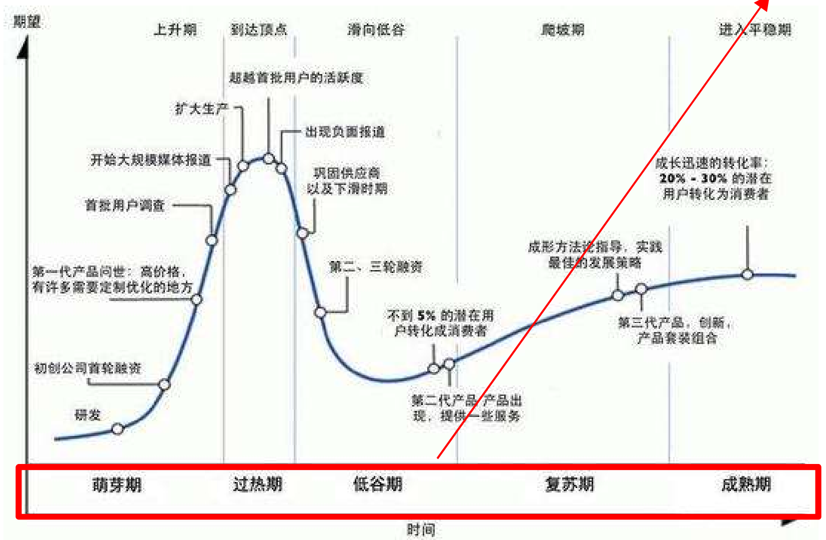
- 熵（Entropy）的应用举例
  - 使用熵进行公司IPO(首次公开募股）预测：衡量公司技术分布

萌芽期
过热期
低谷期
复苏期
成熟期

公司的主题在5个发展阶段上的分布越均衡，说明在处于各项时期的技术都有研发投入，那么它的发展前景就越好。

$$IOP(a) = -\sum_{i=1}^{5} P_i log_2 P_i,$$

$$P_i = \sum_{t=1}^{T} \left(P(c_t|a) \sum_{l \in s_i} P(c_t, d_l) \right)$$



Technology Prospecting for High Tech Companies through Patent Mining (Jin Bo et al, ICDM' 2014)
通过专利挖掘实现高科技公司的技术前景

数据预处理：数据变换-离散化

- 熵（Entropy）的应用举例

$$log\ P(y|x) = log(\hat{y}^y \cdot (1-\hat{y})^{1-y}) = ylog\ \hat{y} + (1-y)log(1-\hat{y})$$

$$\hat{y} = P(y=1|x)$$

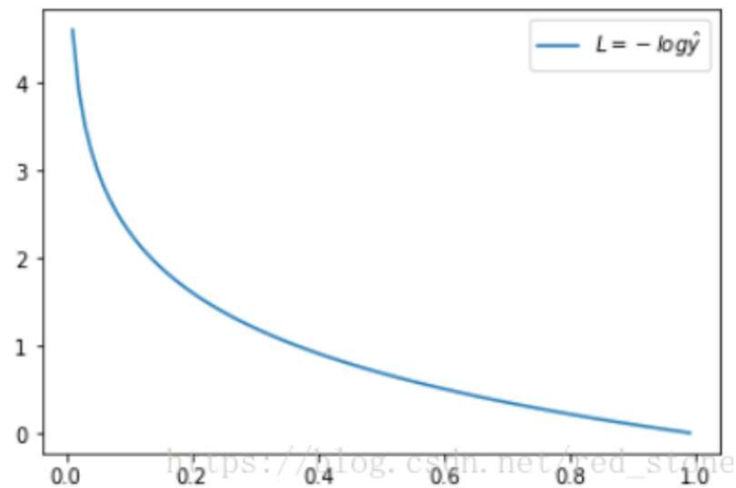
$$1-\hat{y} = P(y=0|x)$$

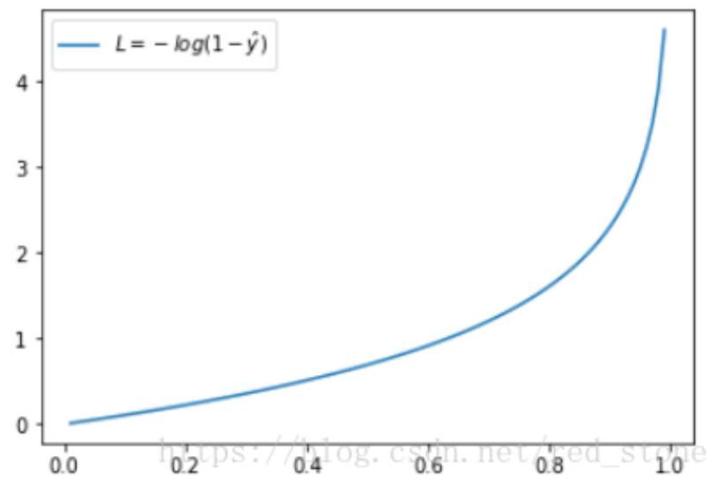
  - 基于交叉熵(Cross Entropy)的机器学习目标函数(Loss Function)

$$-loss = \sum_i (y_i \cdot log(y\_predicted_i) + (1-y_i) \cdot log(1-y\_predicted_i))$$

  - 当y_i=1时，loss优化左半边，　　当y_i=0时，loss优化右半边

L





y_predicted_i

预测输出与y差得越多，L的值越大，即对当前模型的“惩罚”越大，而且是非线性增大，是一种类似指数增长的级别。这是由 log 函数本身的特性所决定的。这样的好处是模型会倾向于让预测输出更接近真实样本标签y。

数据预处理：数据规约

- 为什么需要进行数据规约？
  - 数据仓库中往往存有海量数据
  - 在整个数据集上进行复杂的数据分析与挖掘需要很长的时间

- 数据归约
  - 数据归约可以用来得到数据集的归约表示，它小得多，但可以产生相同的（或几乎相同的）分析结果

数据预处理：数据规约

- 常用的数据归约策略
  - 维度归约，
  - 数值归约，e.g. 使用模型来表示数据

- 用于数据归约的时间不应当超过或“抵消”在归约后的数据上挖掘节省的时间

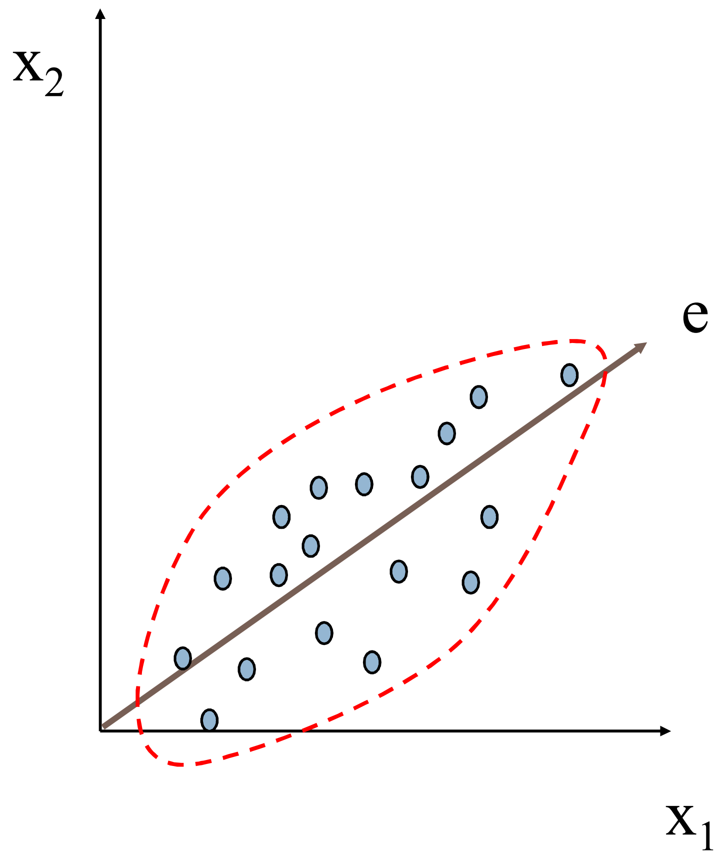
数据预处理：数据规约-维度规约

- 使用数据编码或变换，以便得到原数据的归约或“压缩”表示
- 两种有损的维度归约方法
  - 主成分分析，搜索k个最能代表数据的n维正交向量，其中k小于等于n，这样，原来的数据投影到一个小得多的空间，导致维度归约。
    - 该计算开销低
    - 能够更好的处理稀疏数据
  - 特征子集选择，通过删除不相干的属性或维减少数据集，目标是找出最小属性集。

数据预处理：数据规约-维度规约

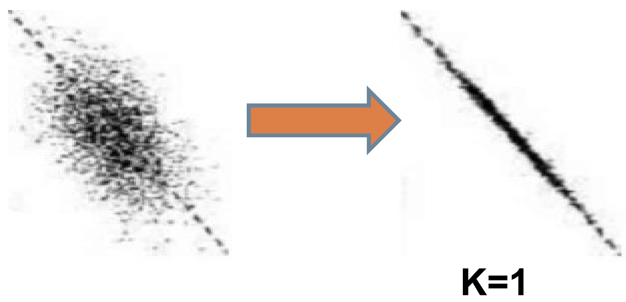
- 使用数据编码或变换，以便得到原数据的归约或“压缩”表示
- 主成分分析

Goal is to find a projection that captures the largest amount of variation in data



数据预处理：数据规约-维度规约

- PCA (Principal Components Analysis)
  - 目的：数据降维、去噪
  - 思想：将原始的高维（如维度为N）数据向一个较低维度（如维度为K）的空间投影，同时使得数据之间的区分度变大。这K维空间的每一个维度的基向量（坐标）就是一个主成分
  - 问题：如何找到这K个主成分
  - 思路：使用方差信息，若在一个方向上发现数据分布的方差越大，则说明该投影方向越能体现数据中的主要信息。该投影方向即应当是一个主成分



数据预处理：数据规约-维度规约

- PCA (Principal Components Analysis)
  - 原理：假设X是一个2*M的数据矩阵(M是数据样本个数)， xi是其中一个2维的数据样本。如果我们想将这些数据X从2维降低到1维：

假设单位列向量 u（2×1），$u^TX=[u^Tx_1, u^Tx_2, ... u^Tx_m]$ u^Tx_i 是每个采样点上的二维数据在单位向量 u 上的投影，由于 X 经过共平均参考处理，所以其均值向量 $\mu = \mathbf{0}$，所以原始观测数据经单位向量 u 投影后的方差

$$VAR（u^TX）= \sum(u^Tx_i)^2 = (u^TX)*(u^TX)^T = u^TXX^Tu = \lambda$$

$u^TXX^Tu = \lambda$ 两边左乘 u 得 $XX^Tu = \lambda u$，显然 u 是 XX^T 的一个特征向量，而 XX^T 是 X 的协方差矩阵， λ 的值的大小表示原始观测数据经在向量 u 的方向上投影值的方差的大小。从而将问题“寻找在投影方向上观测数据分布的方差最大的方向”转变成求原始观测数据 X 的协方差矩阵特征值最大的特征向量的问题。

- 推广：
  - 第K个主成分就是第K大的特征值对应的特征向量
  - 对于原始的N*M（N维M个样本）的数据，原始存储空间是N*M，PCA以后为：K*M（M个K维样本）+N*K(K个特征向量)