数据科学导论
Introduction to Data Science

第二章 数据分析入门

刘 淇
Email: qiliuql@ustc.edu.cn
课程主页：
http://staff.ustc.edu.cn/~qiliuql/DS2017.html

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

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

    - χ^2检验，值越大，两个变量相关的可能性越大

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

数据预处理：数据集成

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

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

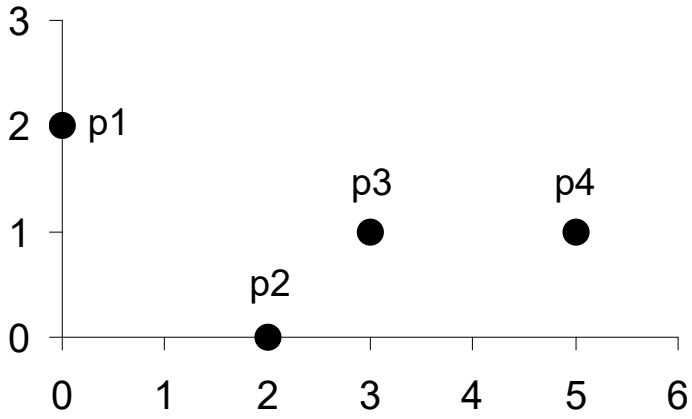
Where n is the number of dimensions (attributes) and p_k and q_k are, respectively, the kth attributes (components) or data objects p and q.

  - 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(名氏距离) is a generalization of Euclidean Distance

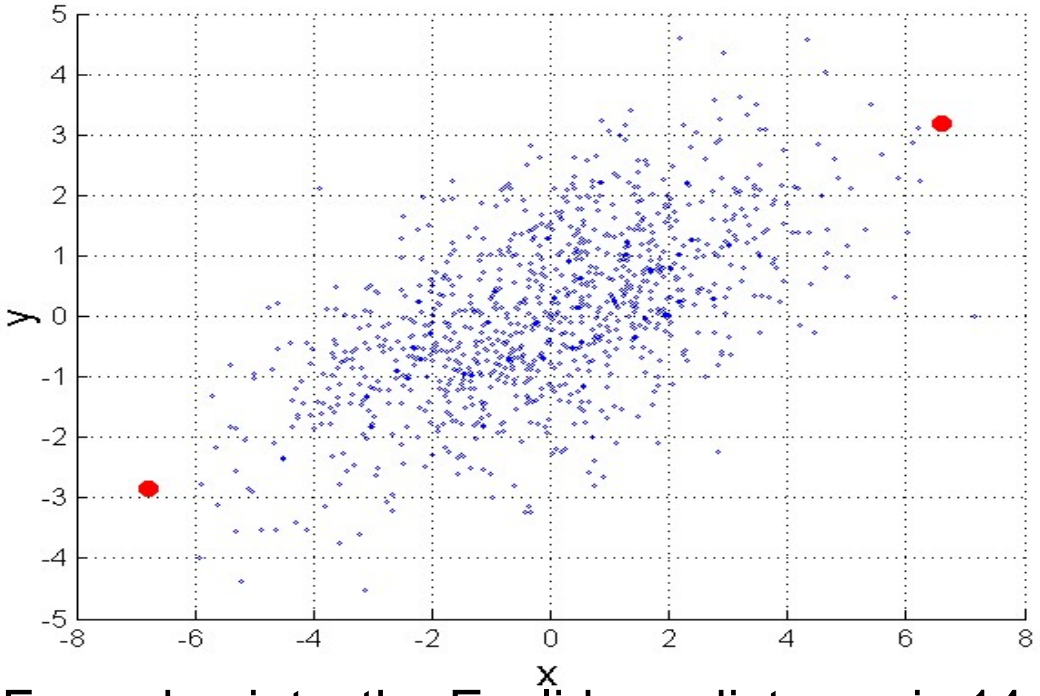
$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

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

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



Σ is the covariance matrix of the input data *X*

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

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.

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

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
- 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)

数据预处理：数据集成

- 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 $\bullet$ indicates vector dot product and $||d||$ is the length of vector d.

- Example:

d_1 = **3 2 0 5 0 0 0 2 0 0**

d_2 = **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
  - **Correlation** 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

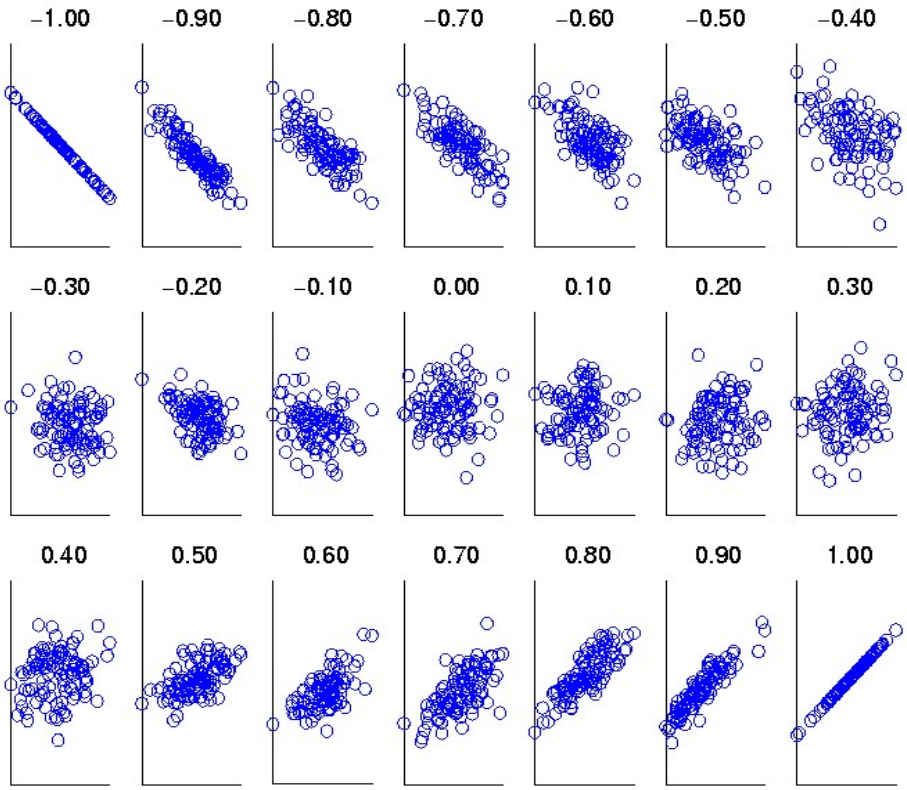
- 数据的距离度量
  - **Correlation** measures the linear relationship between objects
  - To compute correlation, we standardize data objects, p and q, and then take their dot product
  - 可以简单理解为：p和q的协方差/(p的标准差*q的标准差)

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

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.

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
- 思考题：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？

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

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

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

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

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

数据预处理：数据变换

- 数据变换的目的是将数据转换或统一成适合与挖掘的形式。
  - 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－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
  - 最大最小值未知，或者离群点影响较大的时候适用

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

- 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—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

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

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

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

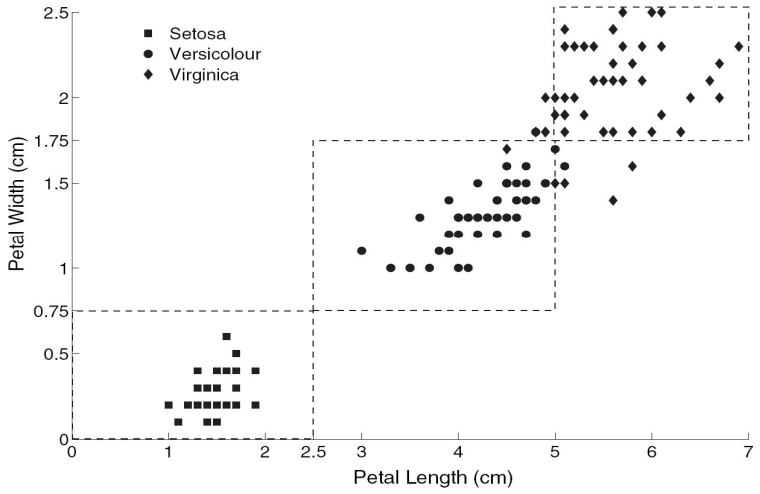
$$v' = \frac{v - \overline{A}}{\sigma_A}$$

  - 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

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

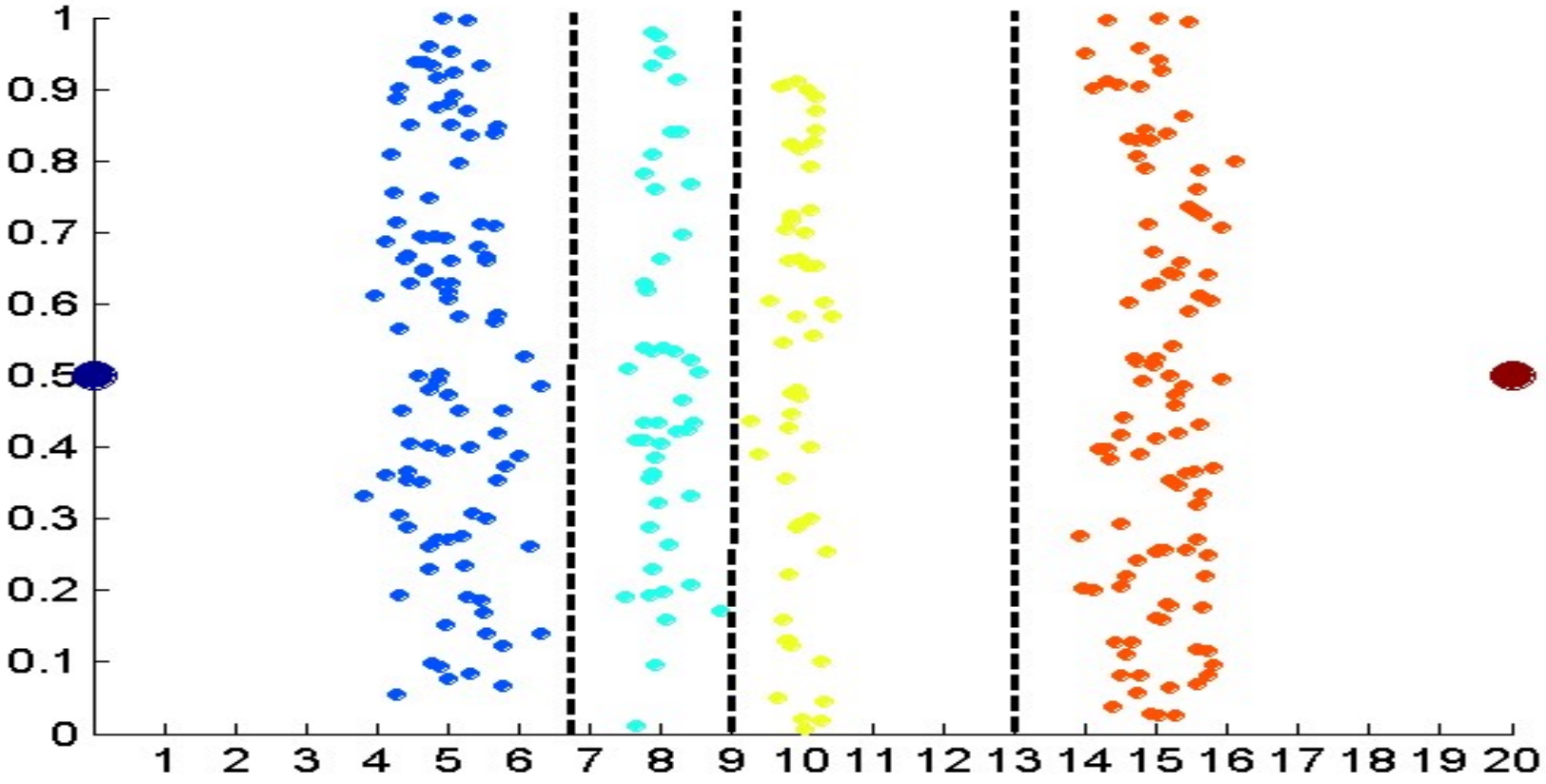
数据预处理：数据变换-离散化

- 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



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

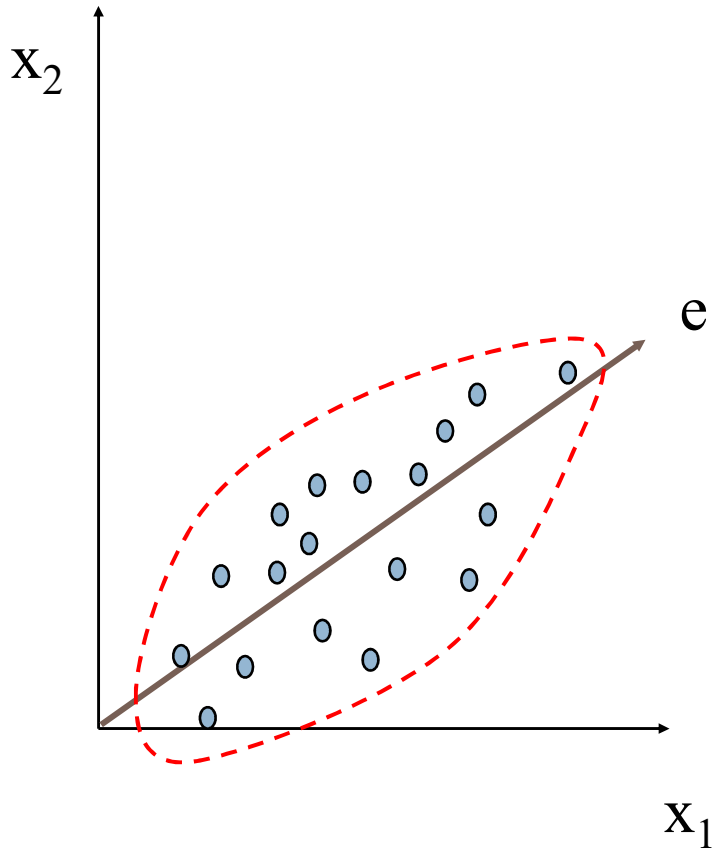
数据预处理：数据规约-维度规约

- 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

K=1

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），$u^TX=[u^Tx_1, u^Tx_2, \ldots u^Tx_m]$ 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 经过其平均参考处理，所以其均值向量 $\mu=0$，所以原始观测数据经单位向量 u 投影后的方差

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

$u^TXX^Tu=\lambda$ 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)

数据预处理：数据规约-维度规约

- **基本计算思路 ---- PCA**

1.对每个样本提取出属性组成一个数字向量

2.对所有样本里的每个属性的取值进行归一化（按属性归一化），以消除不同属性的取值范围等不同带来的影响，得到一个N*M样本矩阵(归一化)；

3.该矩阵乘以该矩阵的逆为协方差矩阵，这个协方差矩阵是可对角化的，对角化后剩下的元素为特征值，每个特征值对应一个特征向量（特征向量要标准化）；

4.选取最大的K个特征值（其中K即为PCA的主元（PC）数，K越少，越降低数据量，但信息丢失也越大，识别效果也越差），将这K个特征值对应的特征向量组成新的矩阵；

5.将新的矩阵转置后乘以样本向量即可得到降维后的数据（这些数据是原数据中相对较为主要的，而数据量一般也远远小于原数据量）。

数据预处理：数据规约-维度规约

省份	GDP X_1	居民消费水平 X_2	固定资产投资 X_3	职工平均工资 X_4	货物周转量 X_5	居民消费价格指数 X_6	商品零售价格指数 X_7	工业总产值 X_8
北京	1394.89	2505	519.01	8144	373.9	117.3	112.6	843.43
天津	920.11	2720	345.46	6501	342.8	115.2	110.6	582.51
河北	2849.52	1258	704.87	4839	2033.3	115.2	115.8	1234.85
山西	1092.48	1250	290.9	4721	717.3	116.9	115.6	697.25
内蒙	832.88	1387	250.23	4134	781.7	117.5	116.8	419.39
辽宁	2793.37	2397	387.99	4911	1371.1	116.1	114	1840.55
吉林	1129.2	1872	320.45	4430	497.4	115.2	114.2	762.47
黑龙江	2014.53	2334	435.73	4145	824.8	116.1	114.3	1240.37
上海	2462.57	5343	996.48	9279	207.4	118.7	113	1642.95
江苏	5155.25	1926	1434.95	5943	1025.5	115.8	114.3	2026.64

数据归一化

三个主成分

第一特征向量 a_1	第二特征向量 a_2	第三特征向量 a_3
0.470641	0.107995	0.19241
0.456708	0.258512	0.109819
0.424712	0.287536	0.19241
−0.31944	0.400931	0.397525
0.312729	−0.40431	0.24505
0.250802	0.498801	−0.24777
0.240481	−0.48868	0.332179
−0.26267	0.167392	0.723351

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
X_1	1.000	.267	.951	.191	.617	−.274	−.264	.874
X_2	.267	1.000	.426	.718	−.151	−.234	−.593	.363
X_3	.951	.426	1.000	.400	431	−.282	−.359	.792
X_4	.191	.718	.400	1.000	−.356	−.134	−.539	.104
X_5	.617	−.151	.431	−.356	1.000	−.255	.022	.659
X_6	−.274	−.234	−.282	−.134	−.255	1.000	.760	−.126
X_7	−.264	−.593	−.359	−.539	.022	.760	1.000	−.192
X_8	.874	.363	.792	−.104	.659	−.126	−.192	1.000

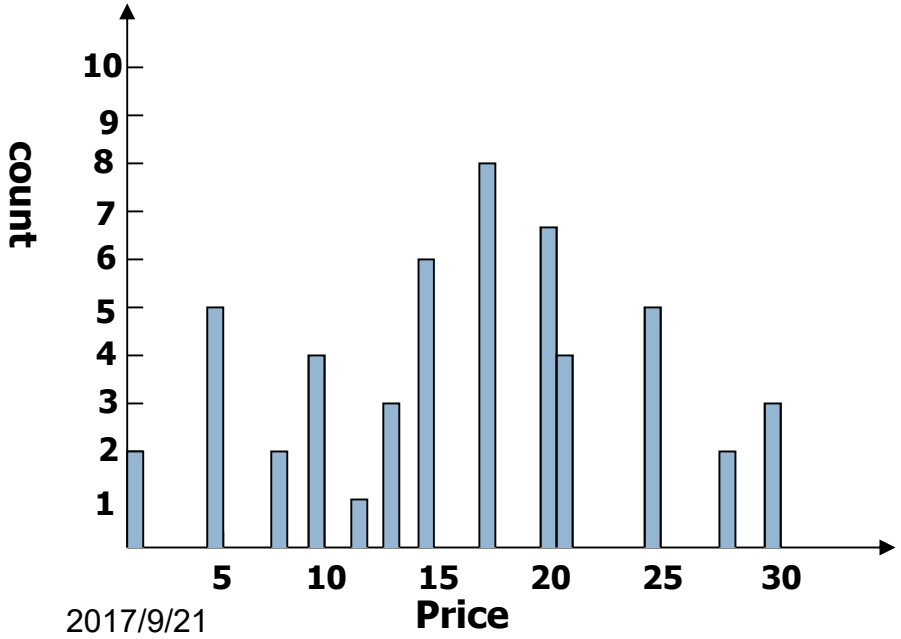
数据预处理：数据规约-维度规约

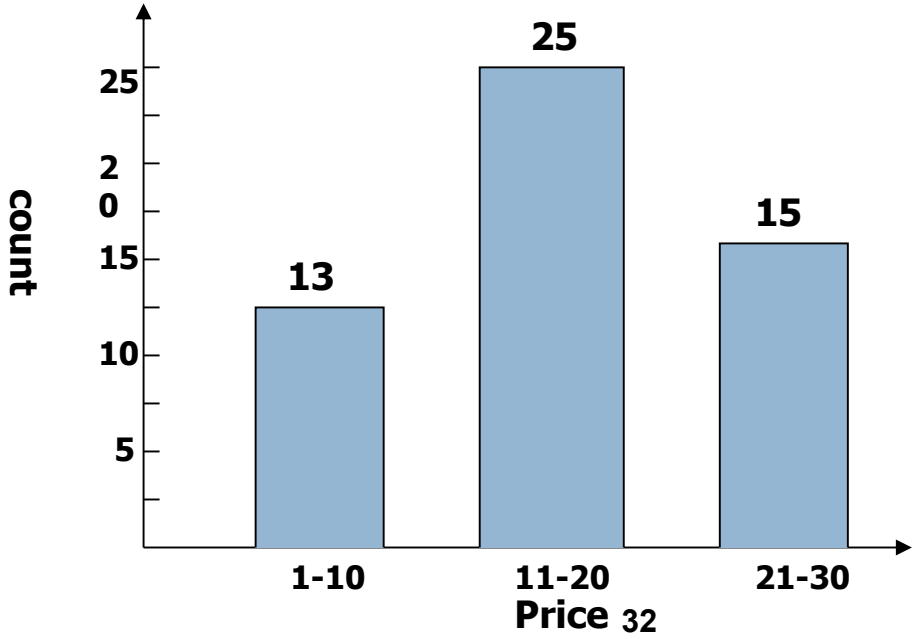
数据预处理：数据规约-数值规约

- 通过选择替代的、较小的数据表示形式来减少数据量
- 有参方法
  - 使用一个参数模型估计数据，最后只要存储参数即可，不用存储数据（除了可能的离群点）
  - 常用方法：线性回归方法；多元回归；对数线性模型；
- 无参方法
  - 不使用模型的方法存储数据
  - 常用方法：直方图，聚类，抽样

直方图

- 类似于分箱技术，是一种流行的数据归约方式
- 将属性值划分为不相交的子集，或“桶”
- 桶安放在水平轴上，而桶的高度（和面积）是该桶所代表的值的平均频率。
- 每个桶只表示单个属性值，则称其为“单桶”。通常，“桶”表示给定属性的一个连续空间





资料推荐

- 数据挖掘导论 第2章：数据，人民邮电出版社
- 数据挖掘原理与算法 第2章， 清华大学出版社
- T.C. Redman *Data Quality: The Field Guide*. January 2001
- I.T.Jolliffe. *Principal Component Analysis*. Springer Verlag, 2nd edition, October 2002.
- *Feature selection algorithms: A survey and experimental evaluation, ICDM 2003*

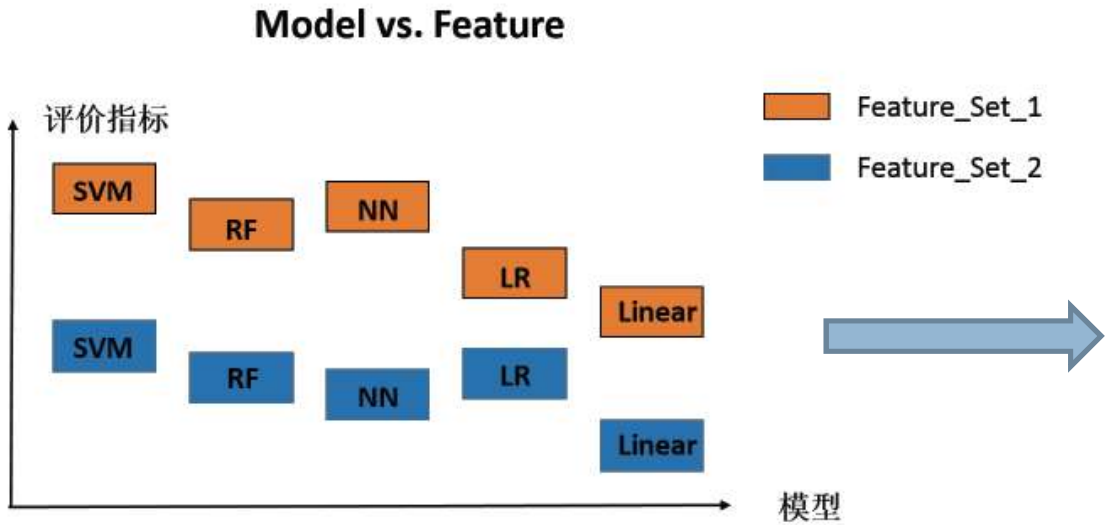
特征工程

- 特征工程定义
- 特征工程的流程
- 特征学习
- 案例学习
- 参考文献

特征工程

- 特征工程是什么？

如何从数据中提取有效的特征，使这些特征能够尽可能的表达原始数据中的信息，使得模型达到更好的效果，就是特征工程(Feature Engineering)所要做的工作。



特征决定了模型的上限

特征工程

特征工程的意义

著名数据科学家Andrew Ng 对特征工程这样描述的：“虽然提取数据特征是非常困难、耗时并且需要相关领域的专家知识，但是机器学习应用的基础就是特征工程”

特征越好，灵活性越强

好的特征能使一般的模型也能获得很好的性能，在不复杂的模型上运行速度 很快，并且容易理解和维护。

特征越好，构建的模型越简单

好的特征不需要花太多的时间去寻找最优参数，降低了模型的复杂度，使模型趋于简单。

特征越好，模型的性能越出色

好的特征能够使模型表现越出色是毫无疑问的，[illegible]就是提升模型的性能。

特征工程的流程

特征工程的流程

1. 对特征进行头脑风暴

深入分析问题，观察数据的基本统计信息，结合问题的相关领域知识和参考其他问题的相关特征工程的方法并应用到自身的问题中来。

2. 特征的设计

人工设计特征、自动提取特征，或者将两者相互结合，得到模型中所使用的特征。

3. 特征的选择

使用不同的特征重要性评分方法或者特征选择方法，对特征的有效性进行分析，选出有效的特征。

4. 评估模型

利用所选择的特征对测试数据进行预测，评估模型的性能。

5. 上线测试

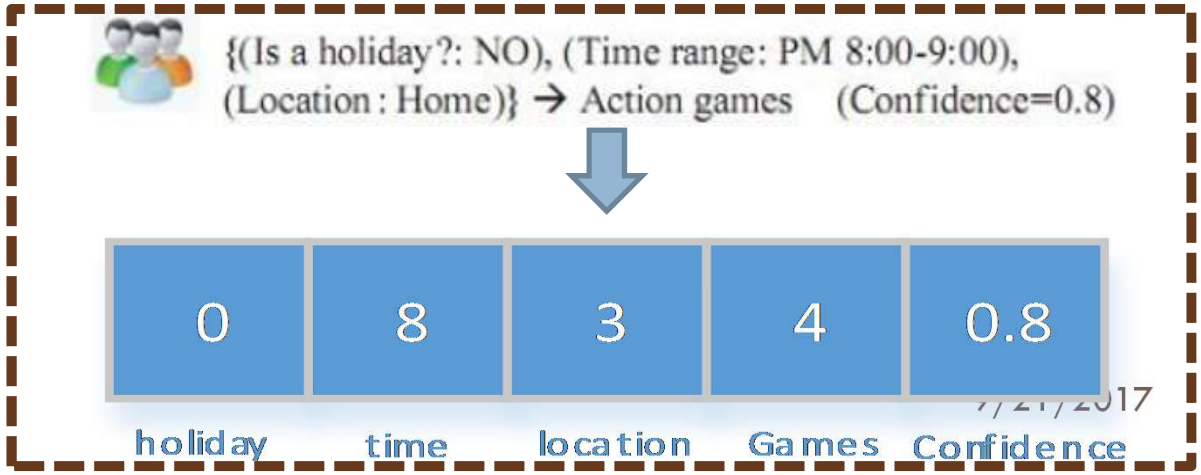
通过在线测试的效果判断特征是否有效，若不能达到要求，则重复2-5 步骤，直到模型的性能达到要求。

特征的设计

- 从原始数据中如何设计特征？
  - 基本特征的提取

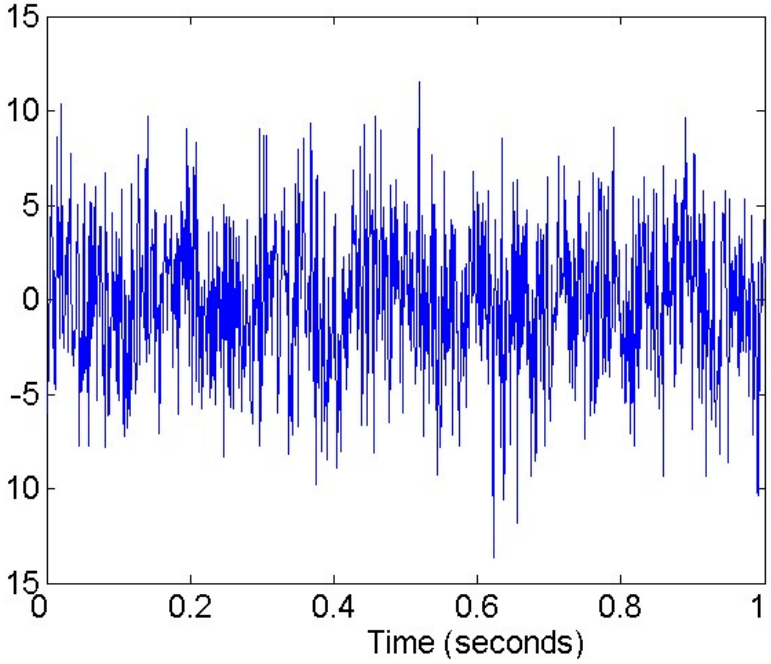
    基本特征的提取过程就是对原始数据进行预处理将其转化成可以使用的数值特征。常见的方法有：数据的**归一化、离散化、缺失值补全**和**数据变化**等。
  - 创建新的特征

    根据对应的领域知识，在基本特征的基础上进行特征之间的比值和交叉变化来构建新的特征。

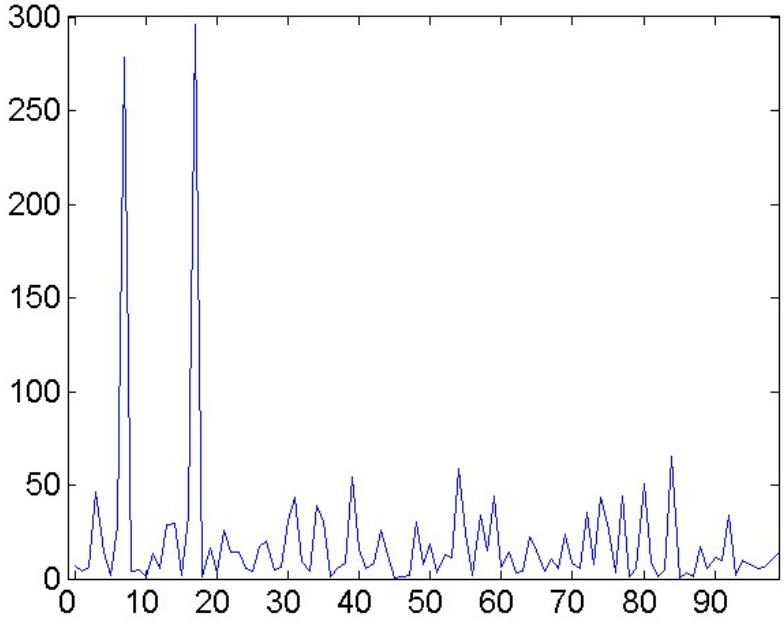


特征的设计

- 从原始数据中如何设计特征？
  - 左图是根据两个Sin函数（分别是每秒7个和17个周期），以及一些噪声数据得到的序列图，右图是由傅立叶变换得到了频率图，可以看出变换后成功得到了两个概率最大的频率7和17（其中纵坐标是振幅，即概率值）



Two Sine Waves(正弦波) + Noise



Frequency

特征的设计

- 举例：第二届“中国高校计算机大赛-大数据挑战赛”
- 赛题描述/数据：http://bdc.saikr.com/vse/bdc/2017
- 简单的说，该赛题的求解目标是利用数据分析将人工的鼠标轨迹和代码生成的鼠标轨迹区分开来。这里的鼠标轨迹是指一种完成一种验证手段——拖动滑块到指定区域时鼠标的轨迹。

  - 原始数据格式：一系列连续点的坐标及其对应时间，目标点的坐标

例如(2,3,4),(2,5,6)(4,3,7) (4,3)，该轨迹中含有三个点的坐标，以(x,y, time)的时间表示，终点坐标为(4,3)

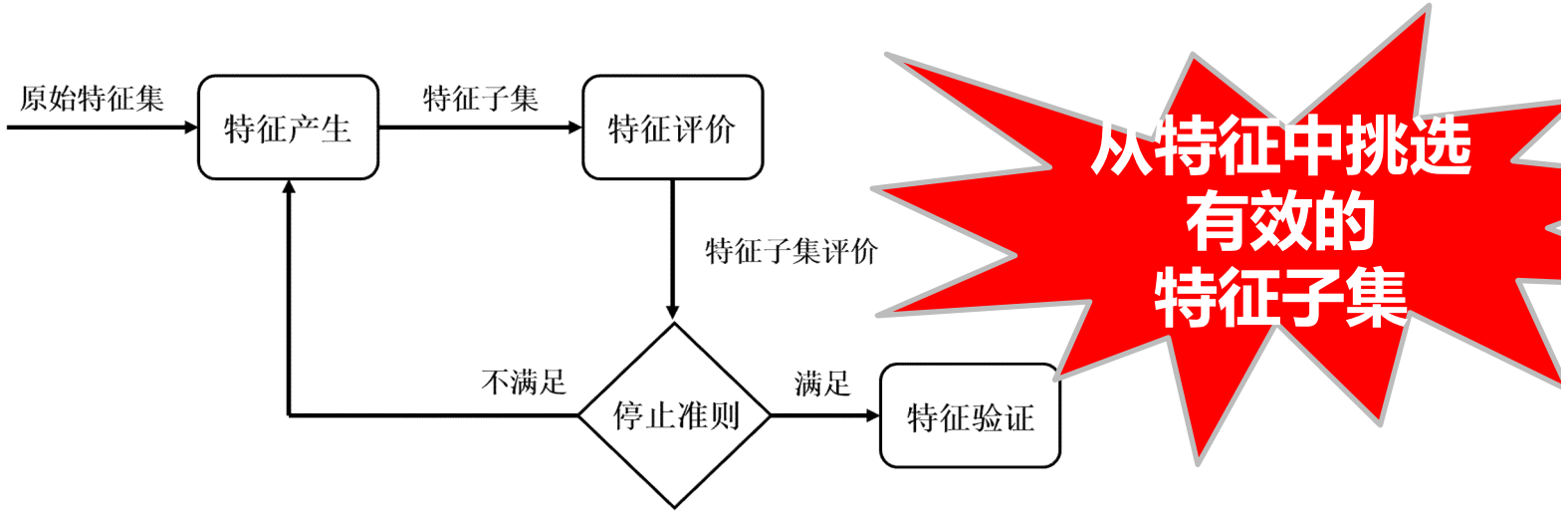
特征的设计

- 从原始数据中如何设计特征？
  - 基本特征的提取
    - 轨迹运动数据的统计值，如运动速度/加速度/角加速度/角速度的均值/极值/最值/中位数 等等
    - 轨迹的描述：运动在x轴方向是否为单向，曲线平滑程度， 等等
  - 创建新的特征
    - 基本特征的简单二元运算， 加/减/乘/除/平方和/和平方/倒数和
    - 运动数据在某一维上的偏导
    - 领域专家知识

特征的选择

- 如何挑选有效的特征？

在实际应用中，特征的数量往往比较多，其中可能会存在不相关的特征。而特征数量越多，分析特征、训练模型所需要的时间就越长，同时容易引起**“维度灾难”**，使得模型更加复杂。特征选择通过剔除不相关的特征或冗余的特征来减少特征数量，从而简化了模型并且提升了模型的泛化能力。



特征选择的过程

特征子集产生过程

- 如何生成特征子集？

特征选择的本质上是一个组合优化的问题，求解组合优化问题最直接的方法就是搜索.根据不用的搜索策略，可以将搜索的算法分为**完全搜索**(Complete), **启发式搜索**(Heuristic) 和**随机搜索**(Random) 三大类。

1. 采用**全局最优搜索策略**的过程产生方法

全局最优搜索策略可以分为穷举搜索与非穷举搜索两类。穷举搜索策略有遍历所有特征和以广度优先搜索的策略，这两种搜索策略都枚举了所有的特征组合，复杂度为2^n。

2. 采用**启发式搜索策略**的过程产生方法

启发式搜索的基本思想是增加关于要解决问题的解某些特征，以便指导搜索向最有希望的方向发展。启发式搜索是搜索是在搜索的最优性和计算量之间做一个折中的搜索策略。

3. 采用**随机算法搜索策略**的产生方法

特征选择本质上是一个组合优化问题，求解这类问题可采用非全局最优目标搜索方法，其实现是依靠带一定智能的随机搜索策略。(如模拟退火，遗传算法等)

特征子集产生过程

- 举例：
  - 对于前面提到的比赛的初步数据特征，我们采用了step-forward的方法来粗糙地选出200个特征。
  - step-forward本质上是一种基于贪心策略的搜索方法，每次从未被选择的特征中挑出能够在测试集上获得最好效果的那个，加入到我们的特征子集中去，直到特征子集数目达到预设值或效果没有明显变化。
  - 思考优缺点：
    - 优点：快，O(nk)的时间复杂度（n为特征全集长度，k为子集最大长度），在大部分情况下可以取得和复杂算法相似的效果
    - 缺点：没有理论的效果保证，最差的情况表现糟糕
    - 弥补措施：多次执行取交集/ 特征分组，以组为粒度选择，增加稳定性/ 加入交叉验证

特征子集评价

如何评价特征子集？

不同的特征选择算法不仅对特征子集评价标准不同，有的还需要结合后续的学习算法模型。因此根据特征选择中子集评价标准和后续算法的结合方式主要分为**过滤式(Filter)**、**封装式(Wrapper)** 和**嵌入式(Embedded)**三种。

1. 过滤式(Filter)评价策略方法

Filter 方法是一种计算效率较高的方法，它**独立于**后续的学习算法模型来分析数据集的固有的属性。通过采用一些基于信息统计的启发式准则来评价特征子集。 启发式的评价函数主要分为四类: **距离度量**、**信息度量**、**依赖性度量**以及**一致性度量**。

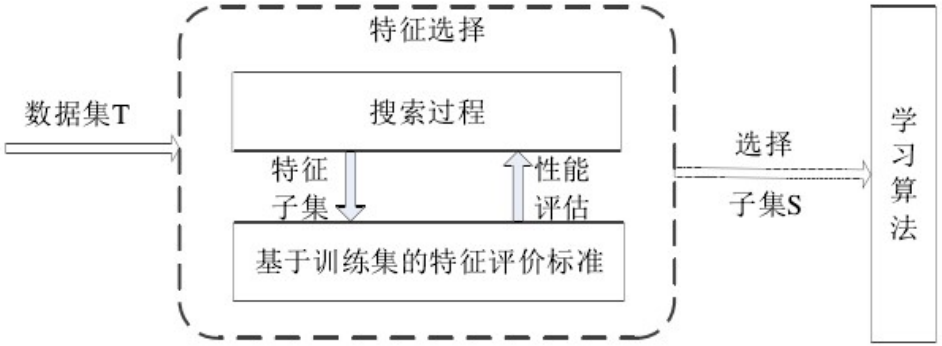


图 2-2 Filter 特征选择

特征子集评价

- 如何评价特征子集？

2. 封装式(Wrapper)评价策略方法

Wrapper选择算法将特征选择作为学习算法一个组成部分，需要**结合后续**的学习算法，并直接将学习算法的分类性能作为特征重要性的评价标准。Wrapper 选择方法直接使用分类器的性能作为评价的标准，选出来的特征子集对分类一定有最好的性能。

相对于Filter 选择方法，Wrapper 方法所选择的特征子集的规模要小得多，有利于关键特征的辨识，模型的分类性能更好。但Wrapper 方法泛化能力较差，当改变学习算法时，需要针对该学习算法重新进行特征选择，算法的计算复杂度高.

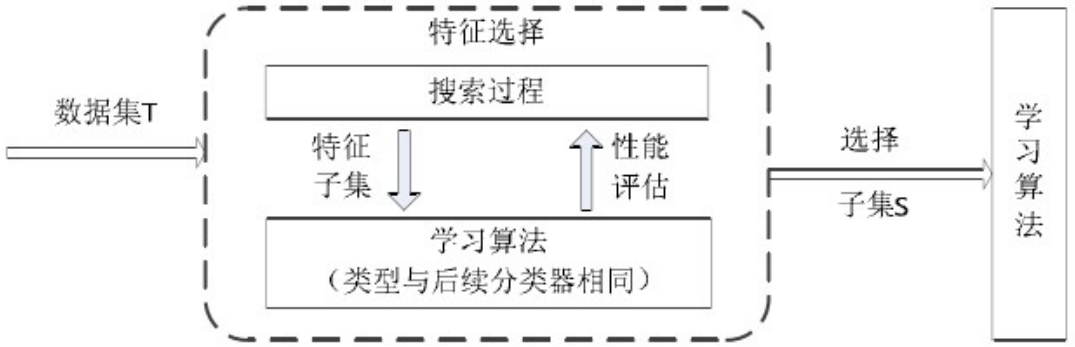


图 2-3 Wrapper 特征选择

特征子集评价

- 如何评价特征子集？

3. 嵌入式(Embedded)评价策略方法

基于Embedded 嵌入式特征选择方法结合了学习算法和特征选择机制去评价学习过程中被考虑的特征。特征选择算法嵌入到学习和分类算法中，也就是特征选择是算法模型中的一部分，算法模型训练和特征选择同时进行，互相结合（即，算法具有自动进行特征选择的功能）。常见的方法有：

1). 带惩罚项的特征选择方法

其基本思想就是在模型损失函数上加上一个惩罚项，模型训练时通过惩罚项来对特征的系数进行惩罚处理，而在特征选择方法中常使用的是L1 正则化项。

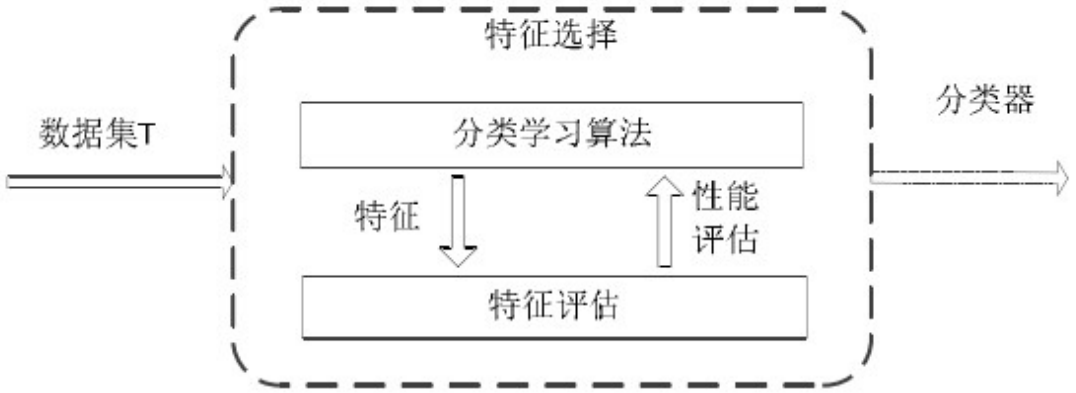


图 2-4 Embedded 特征选择

特征子集评价

- 如何评价特征子集？

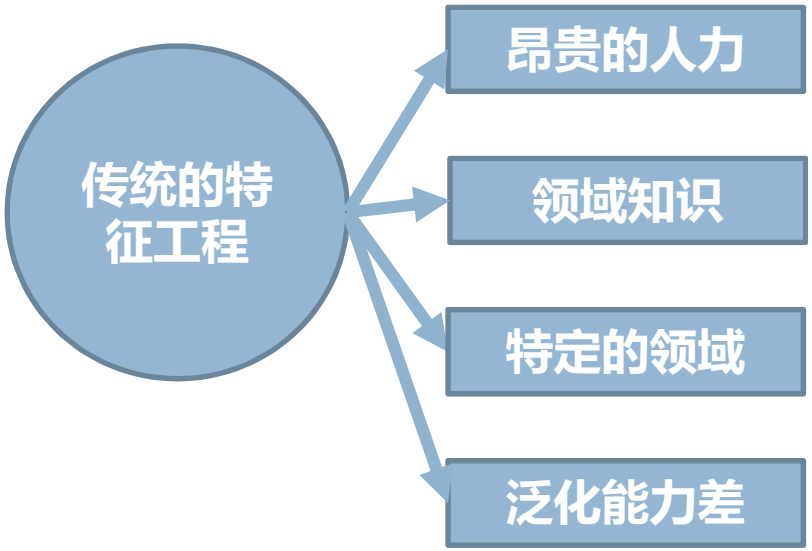
2). 基于树模型的特征选择方法

这些算法在树增长过程的每一步都必须选择一个特征，将样本集划分为纯度更高的子集，而每次选择出的都是使划分效果最佳的特征，所以决策树的生成过程就是特征选择的过程。当决策树完全生成后，每个结点分裂所使用的特征组成的集合就是最后筛选出的特征子集。比如在比赛中经常使用的迭代决策树(GBDT)、随机森林(RF)等算法。

- 举例：
  - 前面初步筛选得到的200维特征，将其输入xgboost(一种高效的梯度提升机器GBM算法)
  - 训练得到特征重要性，也就是分裂树节点时起到的作用权重，自行划分阈值选取特征子集
  - 为了保证不遗漏重要特征，这里不妨将树的深度设高一些

传统特征工程的缺点

- 传统特征工程的缺点

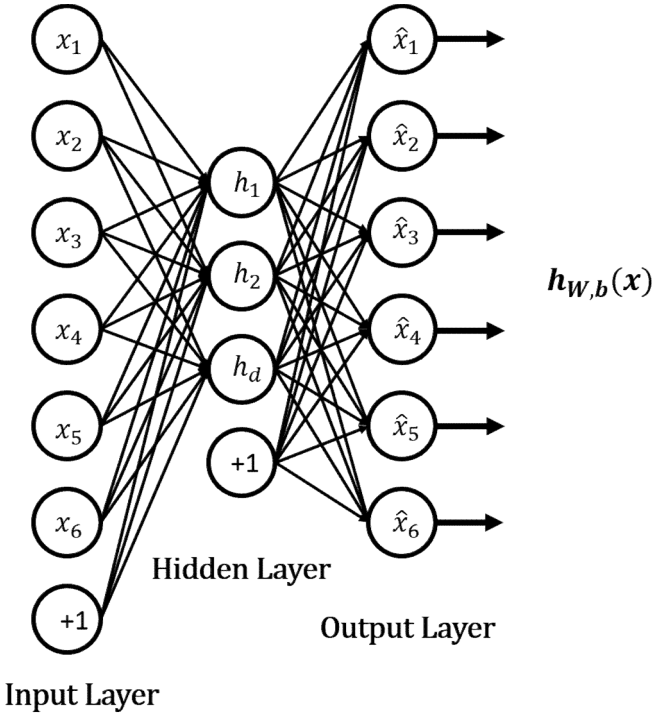


特征学习

- 特征学习

如何从数据中能够自主的学习特征，在这里我们主要介绍在深度学习中常用的三种网络结构。

- 自编码结构(Auto-Encoder)



$h_{W,b}(x)$

将数据的特征X作为Input Layer输入
同样将原始数据特征X′作为Output Layer的输出来重构出原数据。

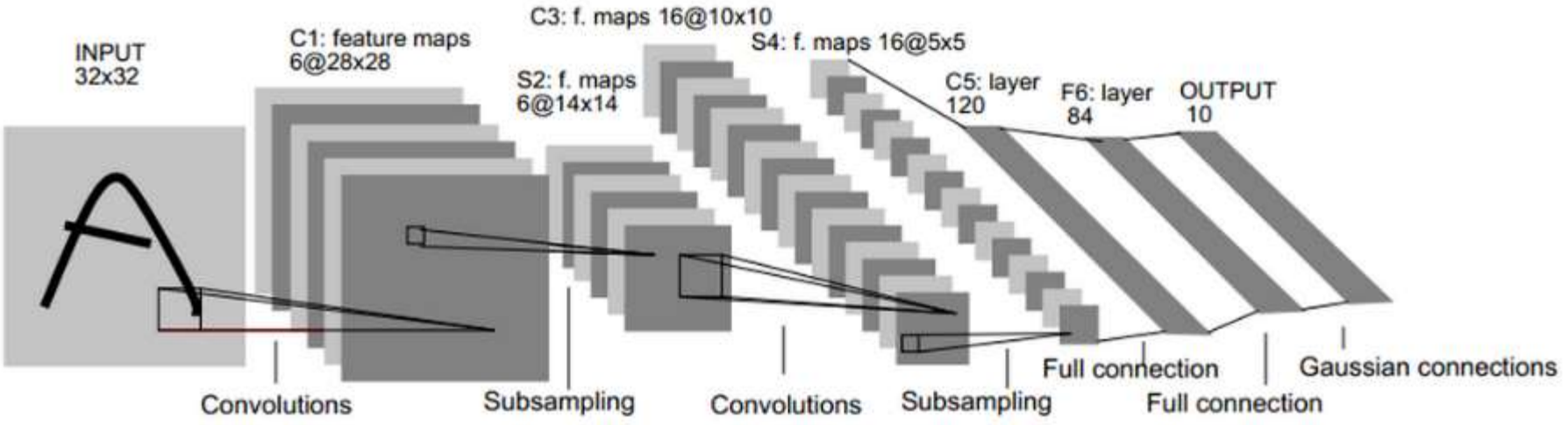
Encoder: $H = f(A * X + b)$

Decoder: $X' = f(A' * H + b')$

将中间的隐含层H的输出作为学习到的数据特征。

特征学习

- 卷积神经网络(CNN): 常用于图像特征提取

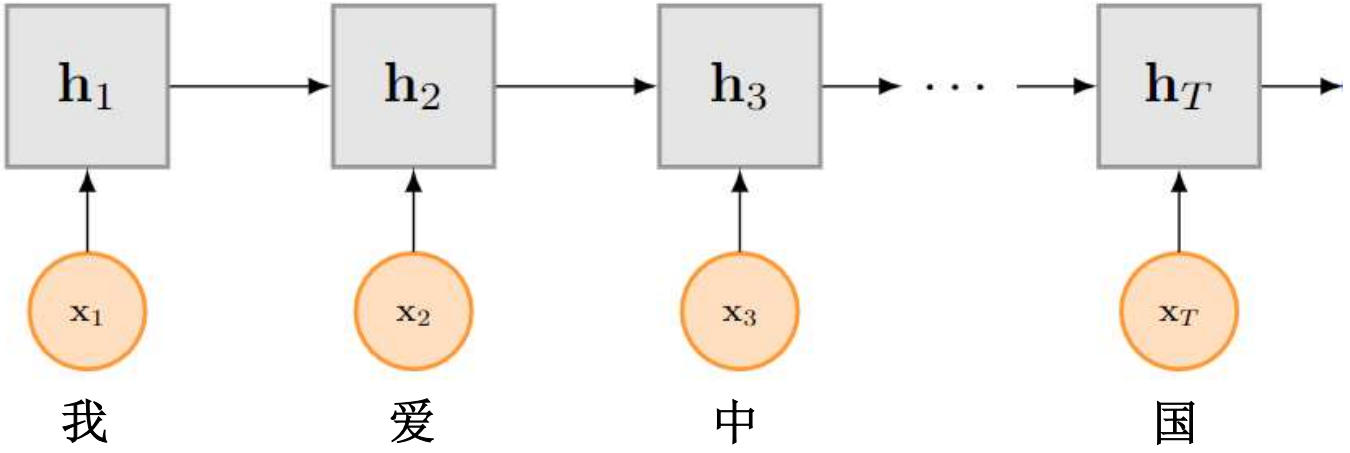


卷积层：通过局部平移，利用不同的卷积核来提取图像中不同的特征
池化层：计算某个区域的特征，提高模型的泛化能力
全连接层：通过多层的神经网络，抽取更高阶的特征。

最终全连接层的输出即为该图像的特征向量表示。

特征学习

- 循环神经网络(RNN): 常用于序列数据的特征提取



将序列中的每个数据依次作为RNN的输入，如上图中的文本数据‘我’、‘爱’、‘中’、‘国’，并将最后一层网络的输出h_T作为最终序列数据的特征向量

课外实践：案例学习

- IRIS(鸢尾花) + sklearn特征工程案例
  - http://www.cnblogs.com/jasonfreak/p/5448385.html
  - 1. 数据集的描述与导入

数据的特征：

花萼长度	花萼宽度	花瓣长度	花瓣宽度

花的类别：

山鸢尾	杂色鸢尾	维吉尼亚鸢尾

```
1 from sklearn.datasets import load_iris
2
3 #导入数据集IRIS
4 iris = load_iris()
5
6 #特征矩阵
7 iris.data
8
9 #目标向量
10 iris.target
```

课外实践：案例学习

- 2. 数据集的预处理

a). 数据的标准化

$$x' = \frac{x - \bar{X}}{S}$$

其中$\bar{X}$为均值，S为标准差

```
1 from sklearn.preprocessing import StandardScaler
2
3 #标准化，返回值为标准化后的数据
4 StandardScaler().fit_transform(iris.data)
```

课外实践：案例学习

- 2. 数据集的预处理

b). 数据的归一化(规则为L2公式如下)：

$$x' = \frac{x}{\sqrt{\sum_j^m x[j]^2}}$$

对特征矩阵的行处理数据，其中m为向量的维度

```
1 from sklearn.preprocessing import Normalizer
2
3 #归一化，返回值为归一化后的数据
4 Normalizer().fit_transform(iris.data)
```

课外实践：案例学习

- 3. 特征的选择

a). Filter(过滤式方法)

使用方差做为Filter的特征评价函数，先要计算各个特征的方差，然后根据阈值选择方差大于阈值的特征。使用 sklearn 通过方差选择法来选择特征的代码如下：

```
1 from sklearn.feature_selection import
                                    VarianceThreshold
2
3 #方差选择法，返回值为特征选择后的数据
4 #参数为方差的阈值threshold
5 VarianceThreshold(threshold=3).fit_transform(iris.
                                    data)
```

课外实践：案例学习

- 3. 特征的选择

b). Wrapper(封装式方法)

在这里我们使用递归特征消除法对一个基模型来进行多轮训练，每轮训练后，消除若干权值系数特征，再基于新的特征集进行下一轮训练，使用 sklearn 通过递归特征消除法来选择特征的代码如下：

```
1 from sklearn.feature_selection import RFE
2 from sklearn.linear_model import
                                  LogisticRegression
3
4 #递归特征消除法，返回特征选择后的数据
5 #参数为基模型estimator
6 #参数为选择的特征个数n_features_to_select
7 RFE(estimator=LogisticRegression(),
  n_features_to_select=2).fit_transform(iris.data,
                                        iris.target)
```

课外实践：案例学习

- 3. 特征的选择

c). Embedded(嵌入式方法)

基于树模型的特征选择法有决策树、随机森林和GBDT 等方法。在这里以 GBDT 来选择特征为例，具体 sklearn 的实现代码如下所示:

```
1 from sklearn.feature_selection import
  SelectFromModel
2 from sklearn.ensemble import
  GradientBoostingClassifier
3
4 #作为基模型的特征选择GBDT
5 SelectFromModel(GradientBoostingClassifier())
  .fit_transform(iris.data, iris.target)
```

参考文献

- 书籍
  - 数据挖掘导论
  - 机器学习
- 论文
  - 《An Introduction to Variable and Feature Selection》
  - 《特征选择常用算法综述》
- 实战经验
  - Sklearn官方文档
  - Kaggle和天池比赛论坛

第二章数据分析入门小结

- 数据采集 Data Acquisition
  - 信息检索
  - 网络爬虫
- 数据预处理 Data Preprocessing
  - 数据清洗
  - 数据集成
  - 数据变换
  - 数据规约
- 特征工程 Feature engineering
  - 特征设计
  - 特征选择