数据挖掘和机器学习

2020年12月13日 9:31

一、引言

1. 数据中的知识发现包含哪几个步骤?
    a. 确定目标应用并且获得一些先关的先验知识
    b. 数据整合：创建一个目标数据集
        i. 50%-70%时间用作数据的整合和准备
        ii. 需要确定初步的属性列表
        iii. 去除或者填充缺失值
        iv. 去除异常值
    c. 选择和预处理
        i. 选择数据挖掘的类型（摘要，分类，回归，关联，聚类）
        ii. 选择合适的算法
        iii. 选择采样方法、考虑采样复杂度和样本分布不均问题
        iv. 降低属性维度和属性值范围
        v. 数据转换（去耦合和标准化数据）时序数据的转化
        vi. 数据可视化
    d. 数据挖掘：寻找目标模型的通用表示
    e. 解释和评估
2. 数据挖掘应用

    目标检测、文本分类、语音识别、传感器数据建模、自动驾驶、专家系统、辅助医疗

二、数据的可行性

1. Hoeffding不等式（N是采样个数，μ是真实概率，v是采样概率）

$$\boldsymbol{P}[|v-\mu|>\epsilon] \leq \mathbf{2}\ \boldsymbol{exp}\ (-\mathbf{2}\epsilon^{\mathbf{2}}\mathbf{N})$$

2. 用Hoeffding不等式说明学习的可行性

    根据不等式，v逼近真实概率不依赖于μ，当采样数量很大时，v可以近似等于μ。

    对于一个确定的假设H，在足够大的数据下错误比例为Ein(h)依概率逼近在整个数据集上的犯错比例Eout(h)

$$\boldsymbol{P}[|\ E_{in}(h)-E_{out}\ (h)|>\epsilon] \leq \mathbf{2}\ \boldsymbol{exp}\ (-\mathbf{2}\epsilon^{\mathbf{2}}\mathbf{N})$$

$$\begin{aligned} & P_D[BAD\ D] \\ = \ & P_D[\boldsymbol{BAD}\ D\ for\ h_1\ \boldsymbol{or}\ \boldsymbol{BAD}\ D\ for\ h_2\ \boldsymbol{or}\ \ldots \boldsymbol{or}\ \boldsymbol{BAD}\ D\ for\ h_M] \\ \leq \ & P_D[\boldsymbol{BAD}\ D\ for\ h_1] + P_D[BAD\ D\ for\ h_2] + \ldots + P_D[BAD\ D\ for\ \ h_M] \\ \leq \ & 2\ exp\ (-2\epsilon^2\mathbf{N}) + 2\ exp\ (-2\epsilon^2\mathbf{N}) + \cdots + 2\ exp\ (-2\epsilon^2\mathbf{N}) \\ = \ & 2\ M\ exp\ (-2\epsilon^2\mathbf{N}) \end{aligned}$$

    备选函数越少，样本数据量越大，样本成为坏样本的概率越小

三、数据和数据预处理

1. 有哪四种不同的属性类型？分别可以进行什么操作?
    a. 标称型数据：众数，熵，列联相关，卡方检验
    b. 序数：中值，百分位，秩相关，游程检验，符号检验
    c. 区间型：均值，标准差，皮尔逊相关，t检验，F检验
    d. 比率数据：几何平均，调和平均，百分比变差
2. 非对称属性
    a. 只有少量非零数据是重要的属性
3. 数据对象之间相似度，相异度计算?
    a. 相异度
        i. 欧式距离
        ii. 马氏距离
        iii. 明可夫斯基距离

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

b. 相似度

i. jaccard系数

J = number of 11 matches / number of not-both-zero attributes values
= $(M_{11}) / (M_{01} + M_{10} + M_{11})$

i. Tanimoto系数

$$T(p,q) = \frac{p \bullet q}{\|p\|^2 + \|q\|^2 - p \bullet q}$$

p*q为向量积

c. 余弦相似度

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

d. 相似度度量的选择：对于连续的，密集的数据使用欧氏距离，对于稀疏数据和非对称数据使用余弦相似度，jaccard系数等

4. 数据预处理的主要任务?
   a. 数据清洗（缺失值处理，噪声数据，离群点，矛盾数据）
   b. 数据集成（合并多个数据库，数据集和文件的数据）*
   c. 数据转换（标准化和聚合）
   d. 数据规约（得到数据集的简化表示）
      i. 直方图、聚类、分层采样
   e. 数据离散化（数据规约的一部分，通常对标称型数据比较重要）
      i. 颜色、职业等非数值型数据、序数、连续值离散
      ii. 优点有降低属性的取值空间、使用区间值代替区间内的真实值，降低数据量，可以应用一些分类算法
      iii. 主要方法
         1) 基于熵的离散化：先把数据集划分为两部分，计算两部分的熵的和，在熵最小的地方划分，然后对熵最大的那部分重复此步骤，直到满足用户需要的数据集个数
         2) 基于卡方分析进行区间合并

5. 处理缺失值的方法?
   a. 直接去除缺失值数据
   b. 填充缺失数据
      i. 通过专家知识填充合理的值
      ii. 使用全局常数、平均值、最大概率的值替代
6. 处理噪声数据的方法
   a. 等宽桶、等深桶
   b. 回归拟合
   c. 聚类
7. 数据集成
   a. 皮尔逊乘积矩相关系数

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \overline{A})(b_i - \overline{B})}{(N-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^{N}(a_i b_i) - N\overline{AB}}{(N-1)\sigma_A\sigma_B}$$

   b. 卡方检验

	Play chess	Not play chess	Sum (row)
Like science fiction	250 (90)	200 (360)	450
Not like science fiction	50 (210)	1000 (840)	1050
Sum (col.)	300	1200	1500

- X^2 (chi-square) calculation (numbers in brackets are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

For freedom (2-1)(2-1)= 1 & significance level 0.001, the X^2 value needed to reject the *Null Hypothesis* is 10.828

c. 卡方区间合并

$$\chi^2=\frac{\left(a-\frac{(a+b)(a+c)}{n}\right)^2}{\frac{(a+b)(a+c)}{n}}+\frac{\left(b-\frac{(a+b)(b+d)}{n}\right)^2}{\frac{(a+b)(b+d)}{n}}+\frac{\left(c-\frac{(c+d)(a+c)}{n}\right)^2}{\frac{(c+d)(a+c)}{n}}+\frac{\left(d-\frac{(c+d)(b+d)}{n}\right)^2}{\frac{(c+d)(b+d)}{n}}$$

$$=\frac{n(ad-bc)^2}{(a+b)(c+d)(a+c)(b+d)}\quad（其中 n=a+b+c+d）$$

d. 3-4-5规则

If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit (msd), partition the range into **3** equi-width intervals

If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into **4** intervals

If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into **5** intervals

四、决策树学习

1. 决策树学习的基本思想

   逼近离散目标函数的方法，学习到的函数由决策树来表示，可以使用if-then来表示

   a. 根节点：测试每个属性，选择最好的划分
   b. 将数据分到指定的子节点上直到停止条件
      i. 所有的数据都有相同的类别
      ii. 所有数据都有相似的属性值
      iii. 早停（比如设置树的最大深度）
   c. 重复第一步

2. 分类错误率，熵，信息增益的概念，如何根据不同度量选择最佳划分

   a. 基尼系数（越小越好）：

   $$GINI(t)=1-\sum_j[p(j\mid t)]^2$$

   $$GINI_{split}=\sum_{i=1}^{k}\frac{n_i}{n}GINI(i)$$

   b. 熵(倾向于选择分类数量很多的属性)（越大越好）

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

   c. 信息增益（越大越好）

   $$GAIN_{split}=Entropy(p)-\left(\sum_{i=1}^{k}\frac{n_i}{n}Entropy(i)\right)$$

   $$GainRATIO_{split}=\frac{GAIN_{Split}}{SplitINFO}\qquad SplitINFO=-\sum_{i=1}^{k}\frac{n_i}{n}\log\frac{n_i}{n}$$

   a. 分类错误率（越小越好）

   $$Error(t)=1-\max_i P(i\mid t)$$

3. 缺失值对决策树有何影响

   a. 影响不纯度的计算
   b. 影响怎么把缺失值实例分配到子节点
   c. 影响如何对一个有缺失值的示例进行分类

4. 给定混淆矩阵，分类效果度量不同指标的含义及计算方法

   a. 准确率

   $$\text{Accuracy}=\frac{a+d}{a+b+c+d}=\frac{TP+TN}{TP+TN+FP+FN}$$

   b. cost：给每个指标乘上一个权重

c. 精度：p=a/(a+c)
d. 召回率：r=a/(a+b)
e. F1函数：F=2a/(2a+b+c)

5. 评价分类器性能的留一法和k折交叉验证
   a. 留一法：无法有效利用数据，训练集和验证集是有关联的
   b. K折交叉验证：准确率是K个平均
6. 过拟合和欠拟合
   a. 欠拟合：模型太简单
   b. 过拟合：异常点影响
      i. 预剪枝：属性值或者类别相同停止分裂，实例数量小于阈值，信息增益小于某个值，卡方检验属性不独立
      ii. 后剪枝：子节点自底向上替换

五、神经网络

1. 神经网络如何计算？有何特点？
   a. 如何计算？感知机学习算法
   b. 特点
      i. 至少含有一个隐藏层的多层神经网络是一种普适近似，可以近似任何目标函数
      ii. ANN可以处理冗余特征
      iii. 对噪声很敏感
      iv. 使用梯度下降法收敛到局部最小值
      v. 训练十分耗时
2. 梯度下降算法
   a. 首先初始化权重矩阵W
   b. 直到收敛，循环：
      i. $\Delta w = 0$
      ii. 对一个实例X={x1,x2....xn}，Y=t
         1) 计算输出O
         2) 更新W，$\Delta w_i = \Delta w_i + \eta(t-o)x_i$
      iii. $w_i = w_i + \Delta w_i$
3. 多层神经网络使用什么算法进行训练
   a. 梯度反向传播算法
   b. 带动量的随机梯度下降法
4. 随机梯度下降法（计算量更小，不容易陷入到局部最优）

六、贝叶斯算法

1. 根据贝叶斯理论，如何计算一个假设h成立之后的后验概率

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

2. 极大后验概率假设和极大似然假设有何区别
   a. 极大后验概率

$$h_{MAP} \equiv \arg\max_{h \in H} P(h \mid D)$$
$$= \arg\max_{h \in H} \frac{P(D \mid h)P(h)}{P(D)}$$
$$= \arg\max_{h \in H} P(D \mid h)P(h)$$

   b. 最大似然（假设h是等概率的）

$$h_{ML} \equiv \arg\max_{h \in H} P(D \mid h)$$

3. 最小描述长度的基本思想
   a. 基本思想是给定一个假设集合H和数据集D，寻找假设h和模型压缩的数据D的最小长度。

Complexity of Model # c

$$h_{MDL} = \arg\min_{h \in H} L_{C_1}(h) + L_{C_2}(D \mid h)$$

4. 贝叶斯最优分类器的基本思想

Key idea: most probable classification of the new instance is obtained by combining the predictions of all hypothesis, weighted by their posterior probabilities

a. 目标是寻找一个判定标准，最小化总体的风险，也就是用最大后验对分类器进行概率加权

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) P(h_i \mid D)$$

5. 朴素贝叶斯分类算法

Naive_Bayes_Learn($examples$)

For each target value v_j

$\hat{P}(v_j) \leftarrow$ estimate $P(v_j)$

For each attribute value a_i of each attribute a

$\hat{P}(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

6. 贝叶斯信念网络的预测和诊断
7. 偏差方差分析

$$E((y - \hat{f}(x))^2) = \sigma^2 + Var[\hat{f}(x)] + (Bias[\hat{f}(x)])^2$$

a. bagging可以降低模型方差

七、基于实例的学习

1. K近邻学习算法
   a. 属性值需要进行标准化
   b. 大的k对噪声不敏感，对离散数据效果好，在数据集较大时效果好
   c. 小的k计算消耗少
2. k近邻学习计算距离时为什么要进行归一化
3. 局部加权线性回归
4. 基于案例的学习和k近邻学习的异同
   a. 相同点：都是懒惰学习，都是通过分析相似的实例，忽略不同的实例
   b. 不同点：基于实例的学习不将实例表示成一个实值点，而是使用丰富的符号表示，提取相似实例的方式更加精巧
5. 懒惰学习和积极学习的区别
   a. 懒惰学习直到查询到来才进行泛化，可以创造局部近似，训练时间短，查询时间长
   b. 积极学习在查询之前就已经泛化，必须创造全局近似，训练时间长，查询时间短
   c. 如果两者有相同的假设空间H，懒惰学习可以表示更复杂的函数

八、集成学习

1. 集成学习的定义
   a. 使用一组模型来获得比单个模型更换性能的算法
2. 集成学习的两个主要问题
   a. 如何生成若干及学习器
   b. 怎么组合这些学习器
3. Stacking/Bagging/Boosting基本思想及其伪代码
   a. stacking

Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
First-level learning algorithms $\mathcal{L}_1, \cdots, \mathcal{L}_T$;
Second-level learning algorithm $\mathcal{L}$.

Process:
for $t = 1, \cdots, T$:
  $h_t = \mathcal{L}_t(D)$ % Train a first-level individual learner h_t by applying the first-level
end; % learning algorithm $\mathcal{L}_t$ to the original data set D
$D' = \emptyset$; % Generate a new data set
for $i = 1, \cdots, m$:
  for $t = 1, \cdots, T$:
    $z_{it} = h_t(\boldsymbol{x}_i)$ % Use h_t to classify the training example $\boldsymbol{x}_i$
  end;
  $D' = D' \cup \{((z_{i1}, z_{i2}, \cdots, z_{iT}), y_i)\}$
end;
$h' = \mathcal{L}(D')$. % Train the second-level learner h' by applying the second-level
% learning algorithm $\mathcal{L}$ to the new data set D'

Output: $H(\boldsymbol{x}) = h'(h_1(\boldsymbol{x}), \cdots, h_T(\boldsymbol{x}))$

b. Bagging

- Getting L samples by bootstraping
- From which we derive:
  - L Classifiers $\in \{-1,1\}$: $c^1, c^2, c^3, ..., c^L$ or
  - L Estimated probabilities $\in [0,1]$: $p^1, p^2, p^3, ..., p^L$
- The aggregate classifier becomes

$$c_{bag}(x) = sign\left(\frac{1}{L}\sum_{b=1}^{L} c^b(x)\right) \text{ or } p_{bag}(x) = \frac{1}{L}\sum_{b=1}^{L} p^b(x)$$

c. Boosting

Input: Instance distribution $\mathcal{D}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

Process:

```
1.  D_1 = D.        % Initialize distribution
2.  for t = 1, ..., T:
3.      h_t = L(D_t);     % Train a weak learner from distribution D_t
4.      ε_t = Pr_{x~D_t,y} I[h_t(x) ≠ y];     % Measure the error of h_t
5.      D_{t+1} = Adjust_Distribution(D_t, ε_t)
6.  end
```

Output: $H(\boldsymbol{x}) = Combine\_Outputs(\{h_t(\boldsymbol{x})\})$

4. 为何集成学习有效
   a. 若果基学习器是准确且非同质的，那么集成之后子分类器会比单个分类器效果好
   b. 集成之后对单个学习器的容错率变高了

九、分类技术

1. 基于规则的分类器有何优点？需要解决什么问题
   a. 优点：
      i. 表达能力几乎等于决策树
      ii. 通常被用来产生更易于解释的描述性模型
      iii. 非常适合处理类分布不平衡的数据集
   b. 问题：
      i. 多个规则被触发，但是指定不同的类(预先确定规则的优先级)
      ii. 没有一个规则满足
2. 序列覆盖算法

Sequential covering algorithm

```
1. Let E be the training records and A be the set of attribute-value pairs, {(A_j, v_j)}
2. Let Y_o be the ordered set of classes {y_1, y_2, ..., y_k}
3. Let R = {} be the initial rule list
4. for each class y ∈ Y_o - {y_k} do
5.    while stopping condition is not met do
6.       r ← Learn-One-Rule (E, A, y)
7.       Remove training records from E that are covered by r
8.       Add r to the bottom of the rule list: R → R ∪ r
9.    end while
10. end for
11. Insert the default rule, {} → y_k, to the bottom of the rule list R
```

3. 支持向量机基本原理
   a. 寻找一个最优分类超平面最大化间隔

十、聚类分析

1. 聚类的定义
   a. 聚合数据成很多簇，簇间相似度低，簇内相似度高
2. 聚类的类型
   a. 层次的和划分的（层次聚类允许簇拥有子簇，划分不允许）
   b. 互斥（每个对象指派到单个簇）、重叠和模糊（每个对象可能在不同的簇中，模糊聚类在不同簇中的权重值之和为1）
   c. 完全的和部分的（完全聚类中每个对象都在一个簇中，部分的不一定在簇中）
3. 簇的类型
   a. 明显分离的簇（每个点到同簇中任意点的距离比奥不同簇中所有点的距离更近）
   b. 基于中心的簇（每个点到其簇中心的距离比到任何簇中心的距离更近）
   c. 基于近邻的簇（每个点到该簇中至少一个点的距离比到不同簇中任意点的距离更近）

d. 基于密度的簇（簇是被低密度区域分开的高密度区域）
e. 概念簇（簇中每个点具有有整个点集到处的某种一般共同性质）
f. 由目标方程描述的簇

4. 层次聚类的两种主要类型
   a. 凝聚的（从单点重发，合并点）
   b. 分裂的（从整体出发，分裂点集合）
5. 计算簇间相似性的单链（MIN）和全链（MAX）方法
   a. 单链（两个簇之间距离最短的两个点的距离）
   b. 全链（两个簇之间距离最长的两个点的距离）
6. k均值和K中心点算法
   a. k均值算法（计算复杂度O(n*K*I*d））
      i. 存在的问题（对离散数据不友好，需要合适的k值，对噪声点敏感，无法发现非凸的簇）
      ii. 处理的方法（预处理和后处理）
         1) 标准化数据，去除离群点
         2) 去除所有表示离群点的簇，将SSE高的簇在进行分割，低sse的簇合并

      1: Select K points as the initial centroids.
      2: **repeat**
      3: Form K clusters by assigning all points to the closest centroid.
      4: Recompute the centroid of each cluster.
      5: **until** The centroids don't change

   b. k中心点算法
      i. 将k均值算法的均值改变成取簇中离中心最近的样本点
7. DBScan算法

   Eps=邻居的最大半径

   Min-Pts=在最大半径内最少点的数量

   - Arbitrary select a point p
   - Retrieve all points density-reachable from p w.r.t. Eps and $MinPts$.
   - If p is a core point, a cluster is formed.
   - If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.
   - Continue the process until all of the points have been processed.
8. 聚类评估
   a. 簇内评价
      i. 凝聚度（SSE）

$$WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

      i. 分离度

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

十一、关联分析

1. 概念：项集，频繁项集，支持度，置信度，极大频繁项集，闭频繁项集

表 6-2 购物篮数据的二元 0/1 表示

TID	面包	牛奶	尿布	啤酒	鸡蛋	可乐
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

   a. 项集：一个或多个项的集合，如上图中的{面包，牛奶}
   b. 事务的宽度：事务中出现项的个数
   c. 频繁项集：支持度大于某个阈值的项集

d. 支持度：事务中包含项集的比例

$$\sigma(X) = |\{t_i \mid X \subseteq t_i, t_i \in T\}|$$

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

e. 置信度：集合x和集合y中的项在一条记录中同时出现的次数/集合x出现的次数

$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

f. 超集：若一个集合S2中的每一个元素都在集合S1中，且集合S1中可能包含S2中没有的元素，则集合S1就是S2的一个超集。S1是S2的超集，则S2是S1的真子集，反之亦然

g. 极大频繁项集：频繁项集的所有超集都是非频繁项集

h. 闭频繁项集：直接超集的支持度计数都不等于他本身的支持度计数

2. Apriori算法

- Let $k = 1$
- Generate frequent itemsets of length 1
- *Repeat* until no new frequent itemsets are identified
  - Generate length $(k+1)$ **candidate** itemsets from length k frequent itemsets
    - If k-itemsets are not frequent, the corresponding $(k+1)$-itemsets cannot be frequent
  - Count the support of each candidate by scanning the DB
  - Eliminate candidates that are infrequent, leaving only those that are frequent

3. FP增长算法

(1) 扫描一次数据集，确定每个项的支持度计数。丢弃非频繁项，而将频繁项按照支持度的递减排序。对于图 6-24 中的数据集，a 是最频繁的项，接下来依次是 b, c, d 和 e。

(2) 算法第二次扫描数据集，构建 FP 树。读入第一个事务$\{a, b\}$之后，创建标记为 a 和 b 的结点。然后形成 null→a→b 路径，对该事务编码。该路径上的所有结点的频度计数为 1。

(3) 读入第二个事务$\{b, c, d\}$之后，为项 b, c 和 d 创建新的结点集。然后，连接结点 null→b→c→d，形成一条代表该事务的路径。该路径上的每个结点的频度计数也等于 1。尽管前两个事务具有一个共同项 b，但是它们的路径不相交，因为这两个事务没有共同的前缀。

(4) 第三个事务$\{a, c, d, e\}$与第一个事务共享一个共同前缀项 a，所以第三个事务的路径 null→a→c→d→e 与第一个事务的路径 null→a→b 部分重叠。因为它们的路径重叠，所以结点 a 的频度计数增加为 2，而新创建的结点 c, d 和 e 的频度计数等于 1。

(5) 继续该过程，直到每个事务都映射到 FP 树的一条路径。读入所有的事务后形成的 FP 树显示在图 6-24 的底部。

4. 关联模式分析

十二、维度约减

1. 过滤方法和包装方法有何区别和优劣
   a. 区别：
      i. 过滤方法：使用评估函数在特征相似度上进行计算（函数依赖于数据的原有特征，有一个隐含的假设是特征越相似，准确率越好）
      ii. 包装方法：评估函数侧重于准确率（不依赖于数据结构，但是计算量大，通常使用学习器的性能函数作为评估函数）
   b. 两种方法的优缺点：
      i. 过滤方法：
         1) 优点：执行速度快，更具有普遍性（不根据某个学习器设置，关注特征本身的相似度）
         2) 缺点：倾向于选择较大的子集
      ii. 包装方法：
         1) 优点：精确度高，概括能力强
         2) 缺点：执行速度慢，缺乏普遍性

2. 五种不同的特征搜索方法，基本思想及其伪代码
   a. 朴素序列特征选择：将M个特征逐个送入评价函数，选择得分最高的N个特征组成的特征子集
   b. 顺序前向选择

1. Start with the empty set $Y_0=\{\varnothing\}$
2. Select the next best feature $x^+ = \underset{x\notin Y_k}{\operatorname{argmax}}\left[J(Y_k + x)\right]$
3. Update $Y_{k+1}=Y_k+x^+$; k=k+1
4. Go to 2

   c. 顺序后向选择

1. Start with the full set $Y_0=X$
2. Remove the worst feature $x^- = \underset{x\in Y_k}{\operatorname{argmax}}\left[J(Y_k - x)\right]$
3. Update $Y_{k+1}=Y_k-x^-$; k=k+1
4. Go to 2

   d. 双向搜索

1. Start SFS with the empty set $Y_F=\{\varnothing\}$
2. Start SBS with the full set $Y_B=X$
3. Select the best feature
   $x^+ = \underset{\substack{x\notin Y_{F_k}\\ x\in Y_{B_k}}}{\operatorname{argmax}}\left[J(Y_{F_k} + x)\right]$
   $Y_{F_{k+1}} = Y_{F_k} + x^+$
   Remove the worst feature
   $x^- = \underset{\substack{x\in Y_{B_k}\\ x\notin Y_{F_{k+1}}}}{\operatorname{argmax}}\left[J(Y_{B_k} - x)\right]$
   $Y_{B_{k+1}} = Y_{B_k} - x^-; \quad k = k+1$
4. Go to 2

   e. 顺序浮动前向选择

1. Start with the empty set $Y=\{\varnothing\}$
2. Select the best feature
   $x^+ = \underset{x\notin Y_k}{\operatorname{argmax}}\left[J(Y_k + x)\right]$
   $Y_k = Y_k + x^+; \quad k = k+1$
3. Select the worst feature*
   $x^- = \underset{x\in Y_k}{\operatorname{argmax}}\left[J(Y_k - x)\right]$
4. If $J(Y_k-x^-)>J(Y_k)$ then
   $Y_{k+1}=Y_k-x$; k=k+1
   go to Step 3
   else
   go to Step 2

3. 维度约减结果评估
   a. 概率距离度量

$$J(F') = \int f(P(F'|C_i), P(C_i))\ dF'$$

   b. 概率依赖度量

$$J(F') = \int f(P(F'|C_i), P(F'))\ dF'$$

   c. 熵度量

   a. 类内距离度量