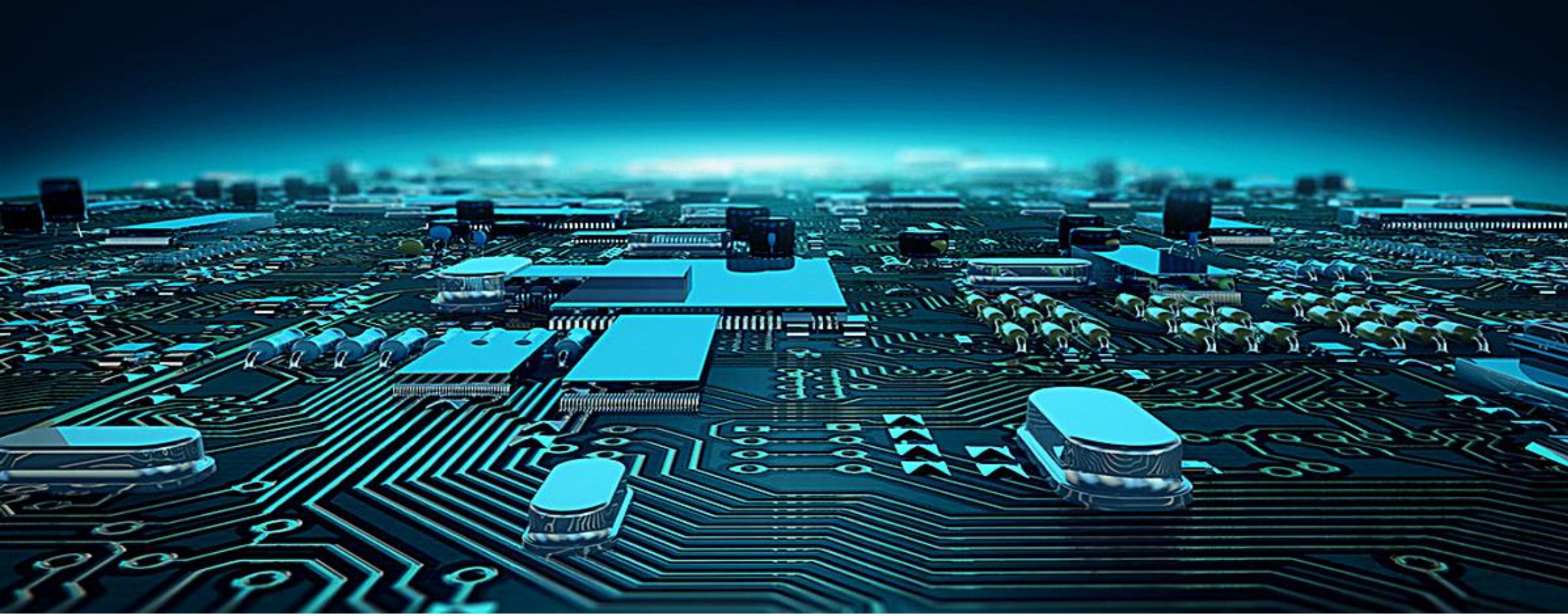
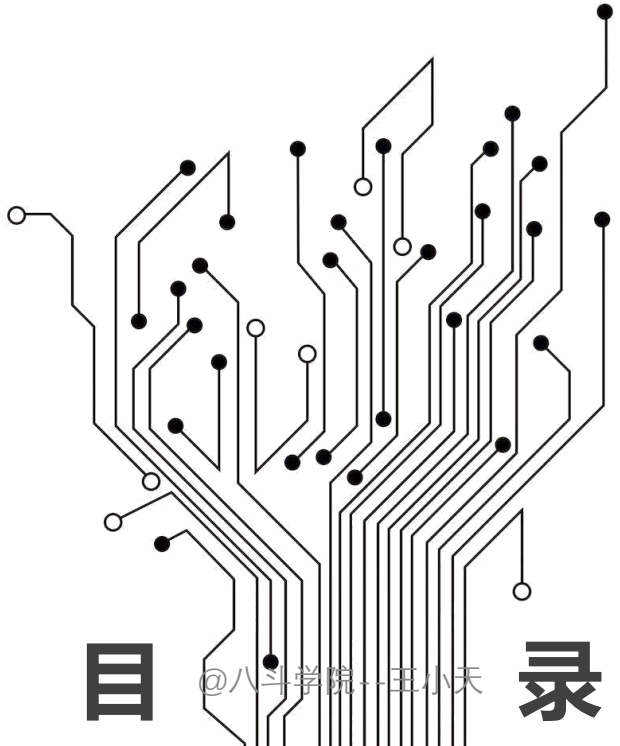


特征选择&特征提取

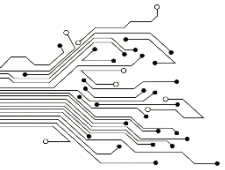
@八斗学院--王小天(Michael)

2021/11/28

1. 特征选择
2. 特征提取
3. PCA



目　录



卷积解决的问题

卷积负责提取图像中的局部特征

特征

在一些实际问题中，我们得到的样本数据都是多个维度的，即一个样本是用多个特征来表征的。比如在预测房价的问题中，影响房价y的因素有房子面积x1、卧室数量x2等，我们得到的样本数据就是（x1,x2）这样一些样本点，这里的x1、x2又被称为特征。

特征选择：为什么要做特征选择?

在现实生活中，一个对象往往具有很多属性（以下称为特征），这些特征大致可以被分成三种主要的类型:

- 相关特征：对于学习任务（例如分类问题）有帮助，可以提升学习算法的效果;
- 无关特征：对于我们的算法没有任何帮助，不会给算法的效果带来任何提升;
- 冗余特征：不会对我们的算法带来新的信息，或者这种特征的信息可以由其他的特征推断出。

但是对于一个特定的学习算法来说，哪一个特征是有效的是未知的。因此，需要从所有特征中选择出对于学习算法有益的相关特征。

进行特征选择的主要目的：

- 降维
- 降低学习任务的难度
- 提升模型的效率

特征选择：什么是特征选择?

定义:

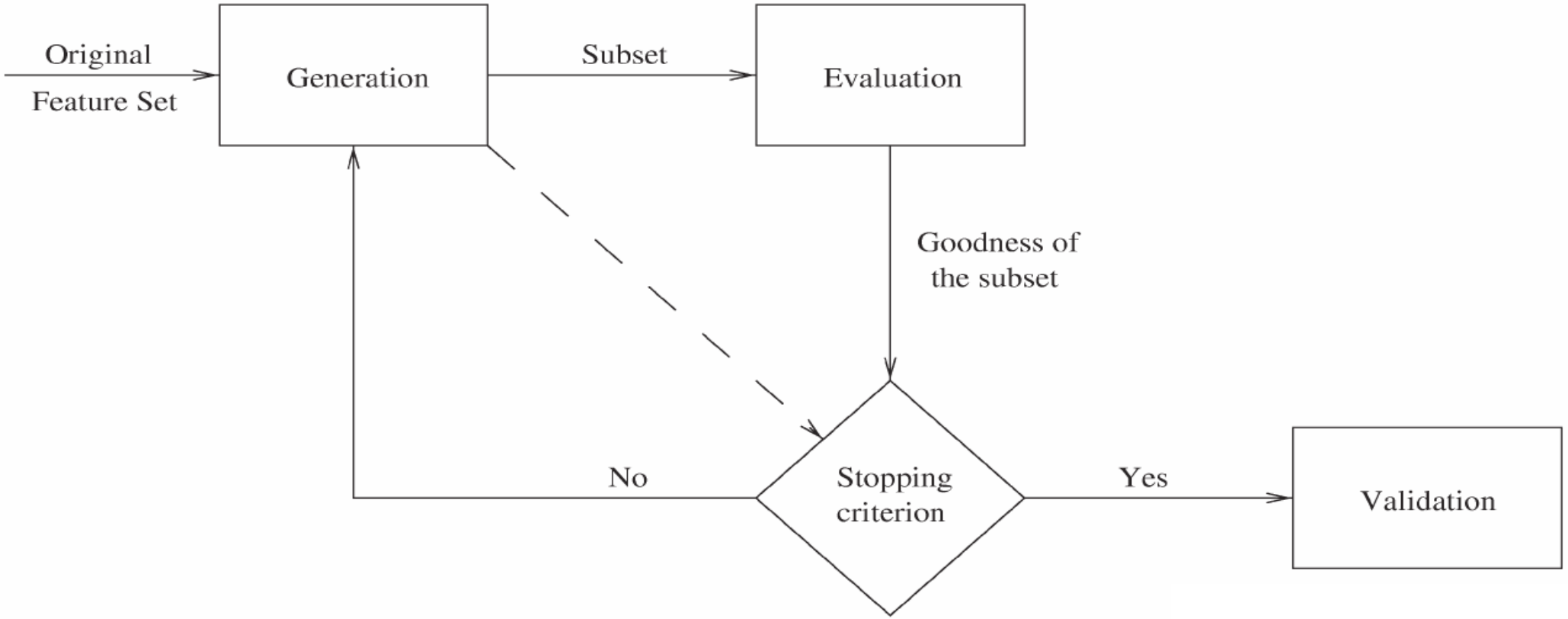
从N个特征中选择其中M（M<=N）个子特征，并且在M个子特征中，准则函数可以达到最优解。

特征选择想要做的是：选择尽可能少的子特征，模型的效果不会显著下降，并且结果的类别分布尽可能的接近真实的类别分布。

特征选择：怎么做特征选择?

特征选择主要包括四个过程:

1. 生成过程: 生成候选的特征子集;
2. 评价函数: 评价特征子集的好坏;
3. 停止条件: 决定什么时候该停止;
4. 验证过程: 特征子集是否有效;



特征选择：生成过程

生成过程是一个搜索过程，这个过程主要有以下三个策略：

1. **完全搜索**：根据评价函数做完全搜索。完全搜索主要有两种：穷举搜索和非穷举搜索；
2. **启发式搜索**：根据一些启发式规则在每次迭代时，决定剩下的特征是应该被选择还是被拒绝。这种方法很简单并且速度很快。
3. **随机搜索**：每次迭代时会设置一些参数，参数的选择会影响特征选择的效果。由于会设置一些参数（例如最大迭代次数）。

特征选择：停止条件

停止条件用来决定迭代过程什么时候停止，生成过程和评价函数可能会对于怎么选择停止条件产生影响。停止条件有以下四种选择：

1. 达到预定义的最大迭代次数；
2. 达到预定义的最大特征数；
3. 增加（删除）任何特征不会产生更好的特征子集；
4. 根据评价函数，产生最优特征子集；

特征选择: 评价函数

评价函数主要用来评价选出的特征子集的好坏，**一个特征子集是最优的往往指相对于特定的评价函数来说的**。评价函数主要用来度量一个特征（或者特征子集）可以区分不同类别的能力。根据具体的评价方法主要有三类：

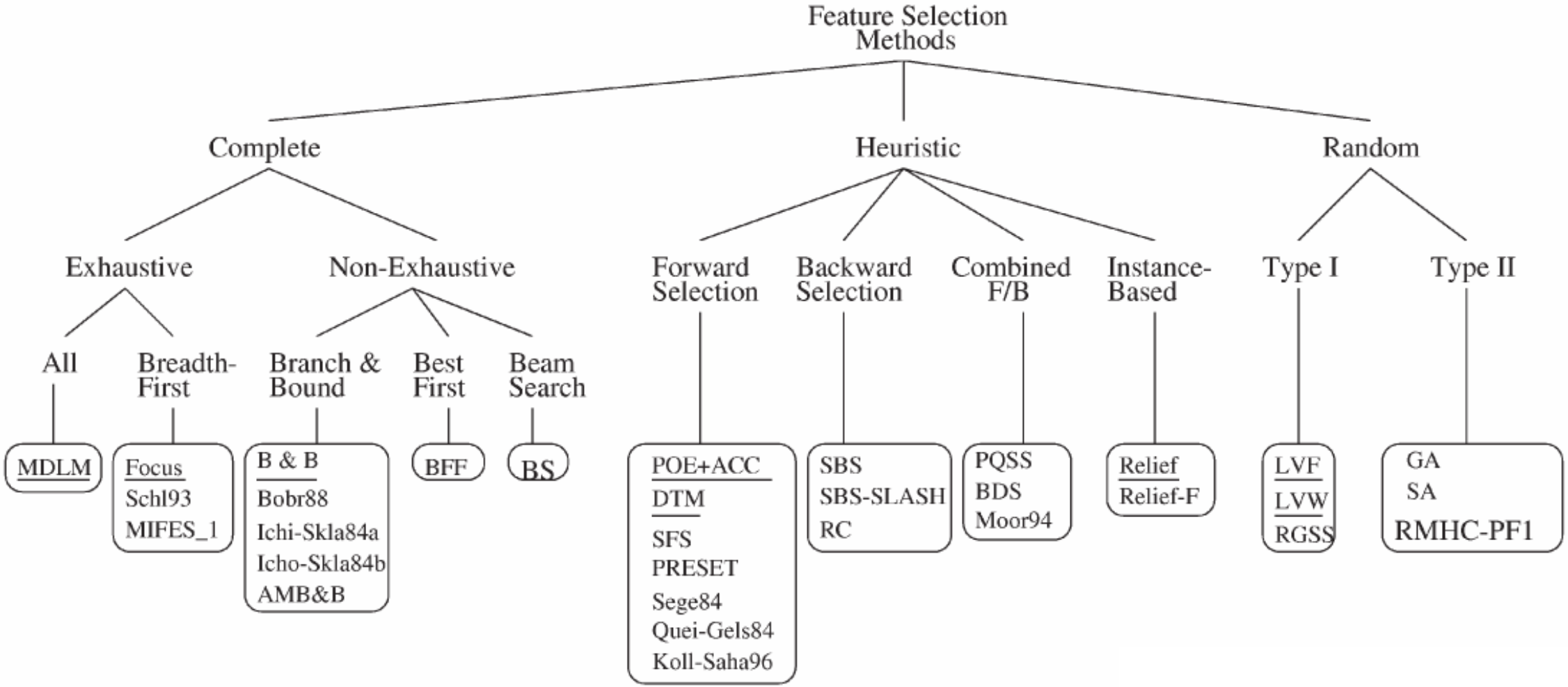
- **过滤式（filter):** 先进行特征选择，然后去训练学习器，所以特征选择的过程与学习器无关。相当于先对于特征进行过滤操作，然后用特征子集来训练分类器。对每一维的特征“打分”，即给每一维的特征赋予权重，这样的权重就代表着该维特征的重要性，然后依据权重排序。
- **包裹式（wrapper）：** 直接把最后要使用的分类器作为特征选择的评价函数，对于特定的分类器选择最优的特征子集。将子集的选择看作是一个搜索寻优问题，生成不同的组合，对组合进行评价，再与其他的组合进行比较。这样就将子集的选择看作是一个优化问题，
- **Filter和Wrapper组合式算法：** 先使用Filter进行特征选择，去掉不相关的特征，降低特征维度；然后利用Wrapper进行特征选择。
- **嵌入式（embedding）：** 把特征选择的过程与分类器学习的过程融合一起，在学习的过程中进行特征选择。其主要思想是：在模型既定的情况下学习出对提高模型准确性最好的属性。这句话并不是很好理解，其实是讲在确定模型的过程中，挑选出那些对模型的训练有重要意义的属性。

特征选择: 评价函数

一般有5种比较常见的评价函数:

1. 距离度量：如果 X 在不同类别中能产生比 Y 大的差异，那么就说明 X 要好于 Y;
2. 信息度量：主要是计算一个特征的信息增益（度量先验不确定性和期望, 后验不确定性之间的差异）;
3. 依赖度量：主要用来度量从一个变量的值预测另一个变量值的能力。最常见的是相关系数：用来发现一个特征和一个类别的相关性。如果 X 和类别的相关性高于 Y与类别的相关性，那么X优于Y。对相关系数做一点改变，用来计算两个特征之间的依赖性，值代表着两个特征之间的冗余度。
4. 一致性度量：对于两个样本，如果它们的类别不同，但是特征值是相同的，那么它们是不一致的；否则是一致的。找到与全集具有同样区分能力的最小子集。严重依赖于特定的训练集和 最小特征偏见（Min-Feature bias）的用法；找到满足可接受的不一致率（用户指定的参数）的最小规模的特征子集。
5. 误分类率度量：主要用于Wrapper式的评价方法中。使用特定的分类器，利用选择的特征子集来预测测试集的类别，用分类器的准确率来作为指标。这种方法准确率很高，但是计算开销较大。

拓展--特征选择算法


Feature Selection Methods
Complete
Heuristic
Random
Exhaustive
Non-Exhaustive
Forward Selection
Backward Selection
Combined F/B
Instance-Based
Type I
Type II
All
Breadth-First
Branch & Bound
Best First
Beam Search
MDLM
Focus
Schl93
MIFES_1
B & B
Bobr88
Ichi-Skla84a
Icho-Skla84b
AMB&B
BFF
BS
POE+ACC
DTM
SFS
PRESET
Sege84
Quei-Gels84
Koll-Saha96
SBS
SBS-SLASH
RC
PQSS
BDS
Moor94
Relief
Relief-F
LVF
LVW
RGSS
GA
SA
RMHC-PF1

拓展--完全搜索:

广度优先搜索（Breadth First Search）：主要采用完全搜索策略和距离度量评价函数。使用广度优先算法遍历所有可能的特征子集，选择出最优的特征子集。

主要采用完全搜索和距离度量。B&B从所有的特征上开始搜索，每次迭代从中去掉一个特征，每次给评价函数的值一个限制条件。因为评价函数满足单调性原理（一个特征子集不会好于所有包含这个特征子集的更大的特征子集），所以如果一个特征使得评价函数的值小于这个限制，那么就删除这个特征。类似于在穷举搜索中进行剪枝。

定向搜索（Beam Search）：主要采用完全搜索策略和误分类率作为评价函数。选择得分最高的特征作为特征子集，把它加入到一个有长度限制的队列中，从头到尾依次是性能最优到最差的特征子集。每次从队列总取得分最高的子集，然后穷举向该子集中加入一个特征后所有的特征集，按照得分把这些子集加入到队列中。

最优优先搜索（Best First Search）：和定位搜索类似，不同点在于不限制队列的长度。

拓展--启发式搜索:

序列前向选择（SFS , Sequential Forward Selection）：使用误分类率作为评价函数。从空集开始搜索，每次把一个特征加入到这个特征子集中，使得评价函数达到最优值。如果候选的特征子集不如上一轮的特征子集，那么停止迭代，并将上一轮的特征子集作为最优的特征选择结果。

广义序列前向选择（GSFS，Generalized Sequential Forward Selection）：该方法是SFS算法的加速算法，它可以一次性向特征集合中加入r个特征。在候选特征中选择一个规模为r的特征子集，使得评价函数取得最优值。

序列后向选择（SBS , Sequential Backward Selection）：把误分类率作为评价函数。从特征的全集开始搜索，每次从特征子集中去掉一个特征，使得评价函数达到最优值。

广义序列后向选择（GSBS，Generalized Sequential Backward Selection）：该方法是SBS的加速，可以一次性的从特征子集中去除一定数量的特征。是实际应用中的快速特征选择算法，性能相对较好。但是有可能消除操作太快，去除掉重要的信息，导致很难找到最优特征子集。

双向搜索（BDS , Bi-directional Search）：分别使用SFS和SBS同时进行搜索，只有当两者达到一个相同的特征子集时才停止搜索。为了保证能够达到一个相同的特征子集，需要满足两个条件:

被SFS选中的特征不能被SBS去除;

被SBS去除的特征就不能SFS选择;

拓展--启发式搜索:

增L去R选择算法（LRS , Plus L Minus R Selection）：

采用误分类率作为评价函数。允许特征选择的过程中进行回溯，这种算法主要有两种形式:

当L>R时，是一种自下而上的方法，从空集开始搜索，每次使用SFS增加L个特征，然后用SBS从中去掉R个特征;

当L<R时，是一种自上而下的算法，从特征的全集开始搜索，每次使用SBS去除其中的R个特征，使用SFS增加L个特征;

序列浮动选择（Sequential Floating Selection）：和增L去R算法类似，只不过序列浮动算法的L和R不是固定的，每次会产生变化，这种算法有两种形式:

序列浮动前向选择（SFFS , Sequential Floating Forward Selection）:从空集开始搜索，每次选择一个特征子集，使得评价函数可以达到最优，然后在选择一个特征子集的子集，把它去掉使得评价函数达到最优;

序列浮动后向选择（SFBS , Sequential Floating Backward Selection：从特征全集开始搜索，每次先去除一个子集，然后在加入一个特征子集。

决策树算法（DTM , Decision Tree Method）：采用信息增益作为评价函数。在训练集中使用C4.5算法，等到决策树充分增长，利用评价函数对决策树进行剪枝。最后，出现在任意一个叶子节点的路径上的所有特征子集的并集就是特征选择的结果。

拓展--随机搜索:

LVF（Las Vegas Filter）：使用一致性度量作为评价函数。使用拉斯维加斯算法随机搜索子集空间，这样可以很快达到最优解。对于每一个候选子集，计算它的不一致性，如果大于阈值，则去除这个子集。否则，如果这个候选子集中的特征数量小于之前最优子集的数量，则该子集作为最优子集。这个方法在有噪声的数据集达到最优解，它是很简单被实现而且保证产生比较好的特征子集。但是在一些特定问题上，它会花费比启发式搜索更多的时间，因为它没有利用到先验知识。

拓展--遗传算法:

使用误分类率作为评价函数。随机产生一批特征子集，然后使用评价函数对于子集进行评分，通过选择、交叉、突变操作产生下一代特征子集，并且得分越高的子集被选中产生下一代的几率越高。经过N代迭代之后，种群中就会形成评价函数值最高的特征子集。它比较依赖于随机性，因为选择、交叉、突变都由一定的几率控制，所以很难复现结果。遗传算法的过程如下:

1. 随机产生初始种群;
2. 在非支配排序后，通过遗传算法的三个算子（选择算子，交叉算子，变异算子）进行变更操作得到第一代种群;
3. 将父代种群与子代种群合并得到大小为N的初始化种群;
4. 对包括N个个体的种群进行快速非支配排序;
5. 对每个非支配层中的个体进行拥挤度计算;
6. 根据非支配关系及个体的拥挤度选取合适的个体组成新的父代种群;
7. 通过遗传算法的基本变更操作产生新的子代种群;
8. 重复 3 到 7 直到满足程序结束的条件（即遗传进化代数）;

特征提取：

特征是什么：常见的特征有边缘、角、区域等。

特征提取：是通过属性间的关系，如组合不同的属性得到新的属性，这样就改变了原来的特征空间。
特征选择：是从原始特征数据集中选择出子集，是一种包含的关系，没有更改原始的特征空间。

目前图像特征的提取主要有两种方法：传统图像特征提取方法 和 深度学习方法。
1. 传统的特征提取方法：基于图像本身的特征进行提取；
2. 深度学习方法：基于样本自动训练出区分图像的特征分类器；

特征选择（feature selection）和特征提取（Feature extraction）都属于**降维（Dimension reduction）**

特征提取：

特征提取的主要方法：主要目的是为了排除信息量小的特征，减少计算量等：

主成分分析（PCA）；

主成分分析PCA：

PCA算法是如何实现的？

简单来说，就是将数据从原始的空间中转换到新的特征空间中，例如原始的空间是三维的(x,y,z)，x、y、z分别是原始空间的三个基，我们可以通过某种方法，用新的坐标系(a,b,c)来表示原始的数据，那么a、b、c就是新的基，它们组成新的特征空间。在新的特征空间中，可能所有的数据在c上的投影都接近于0，即可以忽略，那么我们就可以直接用(a,b)来表示数据，这样数据就从三维的(x,y,z)降到了二维的(a,b)。

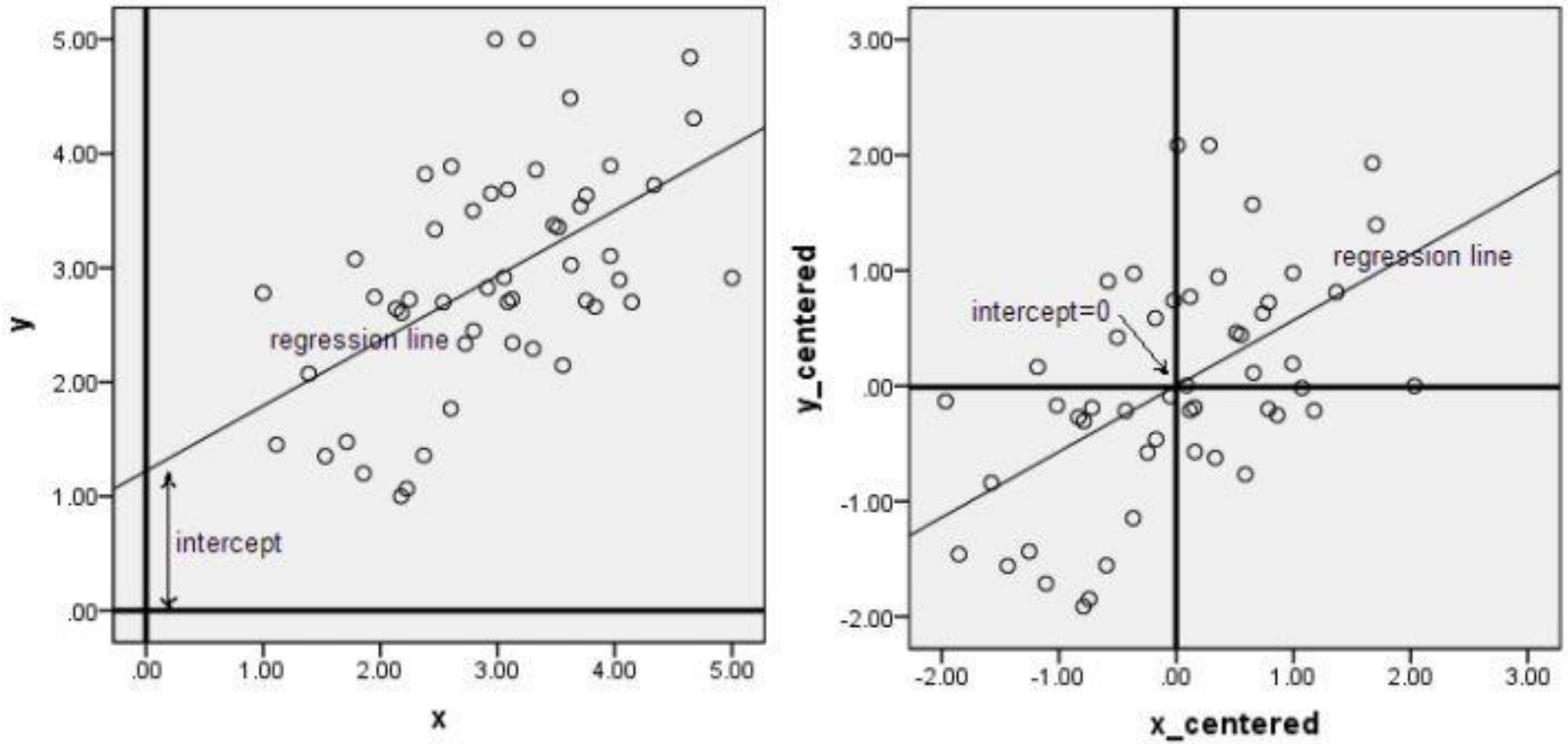
问题是如何求新的基(a,b,c)？

一般步骤是这样的：

1. 对原始数据零均值化（中心化），
2. 求协方差矩阵，
3. 对协方差矩阵求特征向量和特征值，这些特征向量组成了新的特征空间。

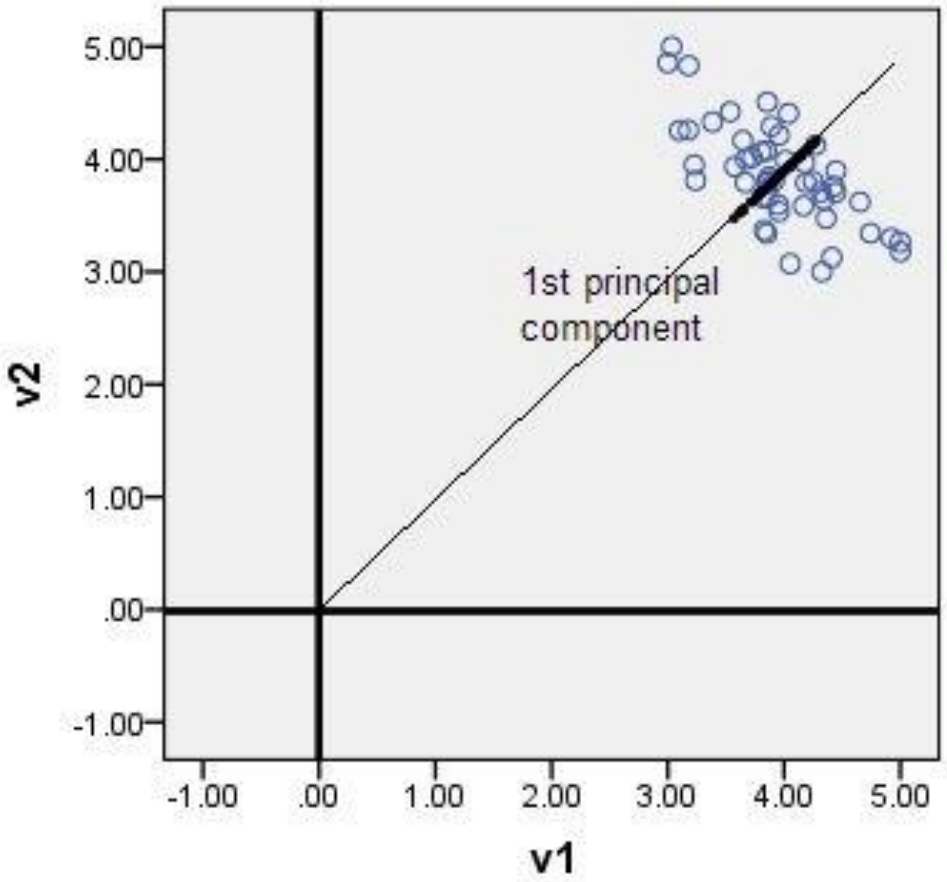
PCA--零均值化（中心化）：

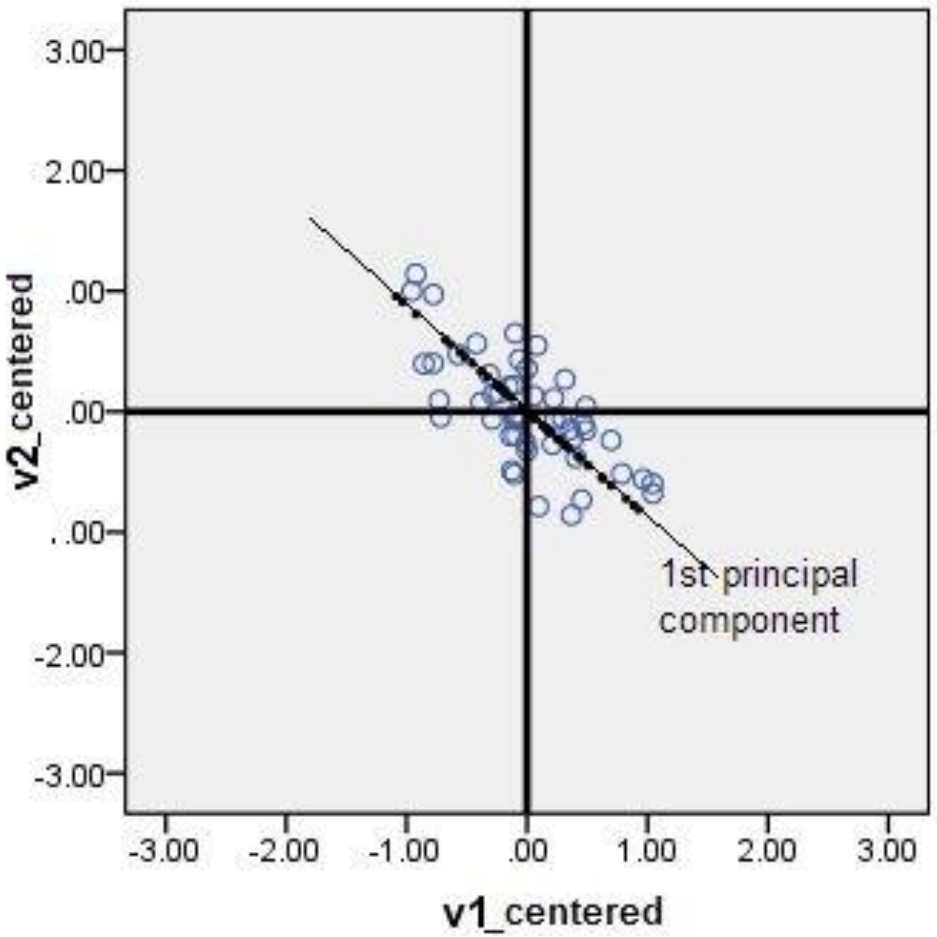
中心化即是指变量减去它的均值，使均值为0。
其实就是一个平移的过程，平移后使得所有数据的中心是(0,0)



PCA--零均值化（中心化）：

只有中心化数据之后，计算得到的方向才能比较好的“概括”原来的数据。
此图形象的表述了，中心化的几何意义，就是将样本集的中心平移到坐标系的原点O上。





PCA--PCA降维的几何意义：

我们对于一组数据，如果它在某一坐标轴上的方差越大，说明坐标点越分散，该属性能够比较好的反映源数据。所以在进行降维的时候，主要目的是找到一个超平面，它能使得数据点的分布方差呈最大，这样数据表现在新的坐标轴上时候已经足够分散了。

方差：

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

PCA算法的优化目标就是：① 降维后同一维度的方差最大

② 不同维度之间的相关性为0

PCA--协方差

协方差就是一种用来度量两个随机变量关系的统计量。
同一元素的协方差就表示该元素的方差，不同元素之间的协方差就表示它们的相关性。

$$cov(X,Y)=\frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{n-1}$$

协方差的性质：

$$s^2=\frac{\sum_{i=1}^{n}(X_i-\bar{X})^2}{n-1}$$

1、Cov(X，Y)=Cov(Y，X)；

2、Cov(aX，bY)=abCov(X，Y)，（a，b是常数）；

3、Cov(X1+X2，Y)=Cov(X1，Y)+Cov(X2，Y)。

PCA--协方差

协方差和方差:

$$cov(X,Y)=\frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{n-1} \qquad s^2=\frac{\sum_{i=1}^{n}(X_i-\bar{X})^2}{n-1}$$

由定义可看出:

$$Cov(X,X)=D(X),\ Cov(Y,Y)=D(Y)$$

协方差衡量了两属性之间的关系，

当$cov(X,Y)>0$时，表示X与Y正相关；

当$cov(X,Y)<0$时，代表X与Y负相关；

当$cov(X,Y)=0$时，代表X与Y不相关。

定义：

$$C = (c_{ij})_{n\times n} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \qquad c_{ij} = Cov(X_i, X_j), i, j = 1, 2, \ldots, n$$

比如，三维(x,y,z)的协方差矩阵：

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

PCA--协方差矩阵

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

协方差矩阵的特点：

- 协方差矩阵计算的是不同维度之间的协方差， 而不是不同样本之间的。
- 样本矩阵的每行是一个样本， 每列为一个维度， 所以我们要按列计算均值。
- 协方差矩阵的对角线就是各个维度上的方差

特别的，如果做了中心化，则协方差矩阵为（**中心化矩阵的协方差矩阵公式，**m为样本个数）：

$$D = \frac{1}{m} Z^T Z$$

PCA—对协方差矩阵求特征值、特征矩阵

A为n阶矩阵，若数λ和n维非0列向量x满足Ax=λx，那么数λ称为A的**特征值**，x称为A的对应于特征值λ的**特征向量**。

式Ax=λx也可写成(A-λE)x=0，E是单位矩阵，并且|A-λE|叫做A 的**特征多项式。**当特征多项式等于0的时候，称为A的特征方程，特征方程是一个齐次线性方程组，**求解特征值的过程其实就是求解特征方程的解。**

对于协方差矩阵A，其特征值λ(可能有多个）计算方法为：

$$|A - \lambda E| = 0$$

行列式 $|A| = ad - bc$, $|A|$是A的二阶行列式

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

PCA—对协方差矩阵求特征值、特征矩阵

$$\mathrm{A}=\begin{pmatrix}3 & 2\\ 1 & 4\end{pmatrix}$$

$$|A-\lambda E|=0$$

$$\det(A-aI)=\det\begin{pmatrix}3-\mathrm{a} & 2\\ 1 & 4-a\end{pmatrix}$$

$$=a^2-7a+10=0$$

该方程有两个根，他们就是特征值：

$$a_1=2,a_2=5$$

代入求得对应的特征向量

$$(A-a_1I)h_1=\begin{pmatrix}1 & 2\\ 1 & 2\end{pmatrix}h_1=0$$

$$\mathrm{h}_1=\begin{pmatrix}2\\ -1\end{pmatrix}$$

$$(A-a_2I)h_2=\begin{pmatrix}-2 & 2\\ 1 & -1\end{pmatrix}h_2=0$$

$$h_2=\begin{pmatrix}1\\ 1\end{pmatrix}$$

PCA—对协方差矩阵求特征值、特征矩阵

对数字图像矩阵做特征值分解，其实是在提取这个图像中的特征，这些提取出来的特征是一个个的向量，即对应着特征向量。而这些特征在图像中到底有多重要，这个重要性则通过特征值来表示。

比如一个100x100的图像矩阵A分解之后，会得到一个100x100的特征向量组成的矩阵Q，以及一个100x100的只有对角线上的元素不为0的矩阵E，这个矩阵E对角线上的元素就是特征值，而且还是按照从大到小排列的（取模，对于单个数来说，其实就是取绝对值），也就是说这个图像A提取出来了100个特征，这100个特征的重要性由100个数字来表示，这100个数字存放在对角矩阵E中。

PCA—对协方差矩阵求特征值、特征矩阵

所以归根结底，特征向量其实反应的是矩阵A本身固有的一些特征，本来一个矩阵就是一个线性变换，当把这个矩阵作用于一个向量的时候，通常情况绝大部分向量都会被这个矩阵A变换得“面目全非”，但是偏偏刚好存在这么一些向量，被矩阵A变换之后居然还能保持原来的样子，于是这些向量就可以作为矩阵的核心代表了。

于是我们可以说：一个变换（即一个矩阵）可以由其特征值和特征向量完全表述，这是因为从数学上看，这个矩阵所有的特征向量组成了这个向量空间的一组基底。而矩阵作为变换的本质其实就是把一个基底下的东西变换到另一个基底表示的空间中。

PCA--对特征值进行排序

* 将特征值按照从大到小的排序，选择其中最大的k个，然后将其对应的k个特征向量分别作为列向量组成特征向量矩阵W_{nxk};
* 计算$X_{new}W$，即将数据集X_{new}投影到选取的特征向量上，这样就得到了我们需要的已经降维的数据集$X_{new}W$。

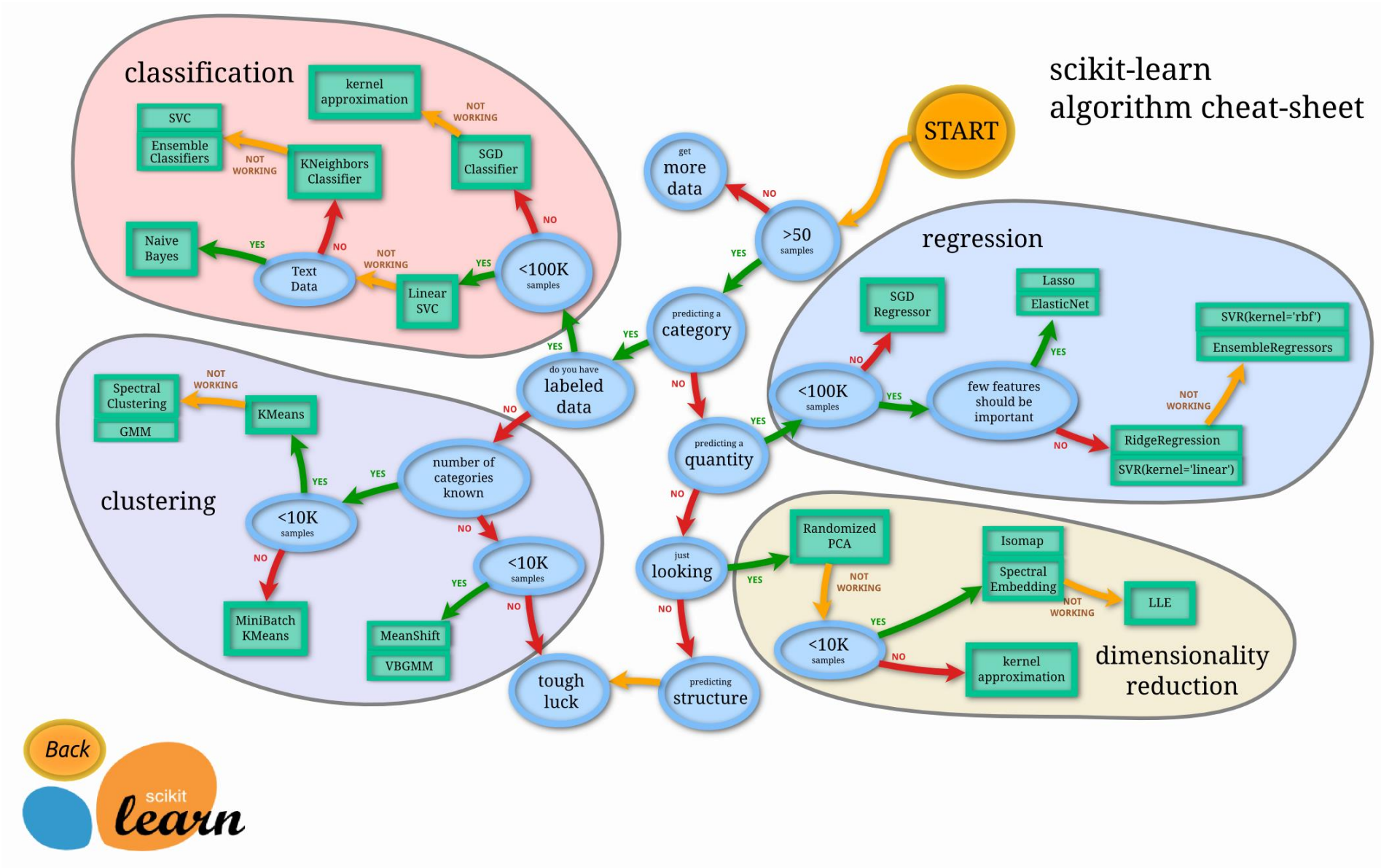
PCA--评价模型的好坏，K值的确定

通过特征值的计算我们可以得到主成分所占的百分比，用来衡量模型的好坏。

对于前k个特征值所保留下的信息量计算方法如下：

$$\eta_k = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{n} \lambda_j} \times 100\%$$

拓展--sklearn库


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
YES
Linear SVC
YES
<100K samples
NO
get more data
NO
>50 samples
YES
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
YES
predicting a category
YES
do you have labeled data
NO
predicting a quantity
YES
NO
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
<10K samples
NO
MiniBatch KMeans
NO
<10K samples
YES
MeanShift
VBGMM
NO
just looking
NO
dimensionality reduction
Randomized PCA
YES
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
tough luck
predicting structure
Back
scikit learn

PCA--鸢尾花实例

我们通过Python的sklearn库来实现鸢尾花数据进行降维，数据本身是4维的，降维后变成2维。

其中样本总数为150，鸢尾花的类别有三种。

萼片长度	萼片宽度	花瓣长度	花瓣宽度	物种
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa
5.4	3.4	1.7	0.2	setosa
5.1	3.7	1.5	0.4	setosa
4.6	3.6	1.0	0.2	setosa
5.1	3.3	1.7	0.5	setosa
4.8	3.4	1.9	0.2	setosa
5.0	3.0	1.6	0.2	setosa
5.0	3.4	1.6	0.4	setosa

PCA算法的优缺点：

优点：

1. 完全无参数限制的。在PCA的计算过程中完全不需要人为的设定参数或是根据任何经验模型对计算进行干预，最后的结果只与数据相关，与用户是独立的。
2. 用PCA技术可以对数据进行降维，同时对新求出的“主元”向量的重要性进行排序，根据需要取前面最重要的部分，将后面的维数省去，可以达到降维从而简化模型或是对数据进行压缩的效果。同时最大程度的保持了原有数据的信息。
3. 各主成分之间正交，可消除原始数据成分间的相互影响。
4. 计算方法简单，易于在计算机上实现。

缺点：

1. 如果用户对观测对象有一定的先验知识，掌握了数据的一些特征，却无法通过参数化等方法对处理过程进行干预，可能会得不到预期的效果，效率也不高。
2. 贡献率小的主成分往往可能含有对样本差异的重要信息。