特征选择

特征的概念

在机器学习中，特征是被观测对象的一个独立可观测的属性或者特点。比如识别水果的种类，需要考虑的特征（属性）有：大小、形状、颜色等。

特征选择的三种算法

特征选择的技术：

1. **包装法（Wrappers methods)**
2. **过滤法（Filters Methods)**
3. **嵌入法（Embedden Methods)**

   包装法**

   基本思想：基于hold-out方法，对于每一个待选的特征子集，都在训练集上训练一遍模型，然后在测试集上根据误差大小选择出特征子集。需要先选定特定算法，通常选用普遍效果较好的算法， 例如Random Forest， SVM， kNN等等。

   优点：面向模型，通常对于选择的模型会有更好的结果

   缺点：计算量大

   用穷举搜索，遍历所有可能的组合达到全局最优，但是计算复杂度为2^n,所以找到最优解是不可能的，但是我们可以采用启发式算法，来加速找到一个满意的解决方案。

   顺序前向搜索

   前向搜索说白了就是，每次增量地从剩余未选中的特征选出一个加入特征集中，待达到阈值或者 n 时，从所有的 F 中选出错误率最小的。过程如下：

   1. 初始化特征集 F 为空。
   2. 扫描 i 从 1 到 n 如果第 i 个特征不在 F 中，那么特征 i 和F 放在一起作为 F_i(即 $F_i = F \cup i$)。 在只使用 F_i 中特征的情况下，利用交叉验证来得到 F_i 的错误率。
   3. 从上步中得到的 n 个 F_i 中选出错误率最小的 F_i ,更新 F 为 F_i 。
   4. 如果 F 中的特征数达到了 n 或者预定的阈值（如果有的话）， 那么输出整个搜索过程中最好的 ； 若没达到，则转到 2，继续扫描。

   递归后向搜索

   既然有增量加，那么也会有增量减，后者称为后向搜索。先将 F 设置为 {1,2,...,n} ， 然后每次删除一个特征，并评价，直到达到阈值或者为空，然后选择最佳的 F 。

   这两种算法时间复杂度都为$O(n^2)$

- **过滤法**

  按照 发散性 或 相关性 对各个特征进行评分，设定阈值或者待选择特征的个数进行筛选

  基本思想：对于每个特征x_i,都计算x_i相对于类别标签y的统计量$J(x_i)$,得到结果，对结果进行排序（$J(x_1), J(x_2), J(x_3)...$）然后输出前k个特征（选择排序在前面的k个）。

  优点：计算量少

  缺点：不是面向模型的

- Person相关系数

  两个变量之间的皮尔逊相关系数定义为两个变量之间的协方差和标准差的商：

  $$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y},$$

  上式定义了**总体**相关系数，常用希腊小写字母 ρ 作为代表符号。估算样本的协方差和标准差，可得到**皮尔逊相关系数**，常用英文小写字母 r 代表：

  $$r = \frac{\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\overline{Y})^2}}.$$

  r 亦可由 (X_i, Y_i) 样本点的标准分数均值估计，得到与上式等价的表达式：

  $$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i-\overline{X}}{\sigma_X}\right)\left(\frac{Y_i-\overline{Y}}{\sigma_Y}\right).$$

  其中 $\frac{X_i-\overline{X}}{\sigma_X}$ 、$\overline{X}$ 及 σ_X 分别是对 X_i 样本的标准分数、样本平均值和样本标准差。

- 卡方检验（x^2-statistic）

  经典的卡方检验是检验**类别型变量**对**类别型变量**的相关性。假设自变量有N种取值，因变量有M种取值，考虑自变量等于i且因变量等于j的样本频数的观察值与期望的差距，构建统计量：

  $x^2 = \sum \frac{(A-E)^2}{E}$

  不难发现，这个统计量的含义简而言之就是自变量对因变量的相关性。

- 互信息和最大信息系数（mutual information）

  互信息和最大信息系数 Mutual information and maximal information coefficient (MIC)

  经典的互信息也是评价**类别型变量**对**类别型变量**的相关性的，互信息公式如下：

  $MI(x_i, y) = \sum_{x_i属于0，1} \sum_{y属于0，1} p(x_i, y) log \frac{p(x_i,y)}{p(x_i)p(y)}$

  当x_i是0/1离散值的时候，这个公式如上。很容易推广到x_i是多个离散值的情况。这里的 $p(x_i, y), p(x_i)$和 $p(y)$都是从训练集上得到的。若问这个 MI 公式如何得来，请看它的 KL 距离（Kullback-Leibler）表述：$MI(x_i, y) = KL(P(x_i, y)||p(x_i)p(y))$也就是说, MI 衡量的是 x_i 和 y 的独立性。如果它俩独立 $P(x_i, y) = p(x_i)p(y)$，那么 KL 距离值为0，也就是x_i和y不相关了，可以去除x_i。相反，如果两者密切相关，那么 MI 值会很大。在对 MI 进行排名后，最后剩余的问题就是如何选择 k 个值（前 k 个 x_i）。(后面将会提到此方法)我们继续使用交叉验证的方法，将 k 从 1 扫描到 n，取评分最高的k。 不过这次复杂度是线性的了。比如，在使用朴素贝叶斯分类文本的时候，词表长度 n 很大。 使用filiter特征选择方法，能够增加分类器精度。

  想把互信息直接用于特征选择其实不是太方便：

  1. 它不属于度量方式，也没有办法归一化，在不同数据及上的结果无法做比较
  2. 对于连续变量的计算不是很方便（X 和 Y 都是集合, x_i, y 都是离散的取值），通常变量需要先离散化，而互信息的结果对离散化的方式很敏感

- **嵌入法**

  原理：分类器执行特征选择作为学习的一部分，比如在目标函数中加入正则项，基于惩罚项的特征选择法 通过L1正则项来选择特征：L1正则方法具有稀疏解的特性，因此天然具备特征选择的特性。

欠拟合和过拟合的概念以及相应的解决措施

一、欠拟合

训练误差和验证误差都很大，这种情况称为欠拟合。出现欠拟合的原因是模型尚未学习到数据的真实结构。因此，模拟在训练集和验证集上的性能都很差。

解决办法：
1、做特征工程，添加很多的特征项。如果欠拟合是由于特征项不够，没有足够的信息支持模型做判断。

2、增加模型复杂度。如果模型太简单，不能够应对复杂的任务。可以使用更复杂的模型，减小正则化系数。具体来说可以使用核函数，集成学习方法。

3、 集成学习方法boosting（如GBDT）能有效解决high bias

二、过拟合

模型在训练集上表现很好，但是在验证集上却不能保持准确，也就是模型泛化能力很差。这种情况很可能是模型过拟合。

造成原因主要有以下几种：
1、训练数据集样本单一，样本不足。如果训练样本只有负样本，然后那生成的模型去预测正样本，这肯定预测不准。所以训练样本要尽可能的全面，覆盖所有的数据类型。

2、训练数据中噪声干扰过大。噪声指训练数据中的干扰数据。过多的干扰会导致记录了很多噪声特征，忽略了真实输入和输出之间的关系。

3、 模型过于复杂。模型太复杂，已经能够死记硬背记录下了训练数据的信息，但是遇到没有见过的数据的时候不能够变通，泛化能力太差。我们希望模型对不同的模型都有稳定的输出。模型太复杂是过拟合的重要因素。

解决办法：
1、在训练和建立模型的时候，从相对简单的模型开始，不要一开始就把特征做的非常多，模型参数跳的非常复杂。

2、增加样本，要覆盖全部的数据类型。数据经过清洗之后再进行模型训练，防止噪声数据干扰模型。

3、正则化。在模型算法中添加惩罚函数来防止过拟合。常见的有L1，L2正则化。

4 集成学习方法bagging(如随机森林）能有效防止过拟合。

5、减少特征个数(不太推荐)
注意：降维不能解决过拟合。降维只是减小了特征的维度，并没有减小特征所有的信息。

在机器学习上核心挑战是我们必须在训练的模型上，在一个新的输入上表现很好。

表现很好的能力我们称为泛化，我们通常计算在测试集上的表现来估计这个模型的泛化误差

通常：

1. 我们对训练集进行抽样，
2. 然后用他来选择参数来减少训练误差
3. 然后用测试集来评估模型

在这个过程中

期望的测试误差>=训练误差

决定模型表现好坏的因素是它的能力：

1. 使训练误差减小
2. 使训练误差和测试误差之间的差距减小

欠拟合：发生在模型不能得到足够低的训练误差

过拟合：发生在训练误差和测试误差的差距很大

我们可以通过改变模型的容量来控制模型是否更容易过拟合或欠拟合。

非正式地说，一个模型的能力是它适应多种功能的能力。

•低容量的模型可能难以拟合训练集。

•具有高容量的模型可以通过记忆训练集的属性来过拟合，这些属性在测试集上不能很好地服务于它们。

控制学习算法容量的一种方法是选择它的假设空间，即允许学习算法选择作为解的函数集。

机器学习算法通常会表现得最好，当它们的能力在以下方面是适当的:•它们需要执行的任务的真正复杂性•它们提供的训练数据量。

欠拟合和过拟合•容量不足的模型无法解决复杂的任务。

•具有高容量的模型可以解决复杂的任务，但是当它们的容量高于解决当前任务所需时，它们可能会过拟合。

虽然简单的函数更容易泛化。

•仍然需要选择一个足够复杂的假设来达到低的训练误差。

欠拟合与过拟合容量与误差的典型关系。

•随着模型容量的增加，训练误差逐渐减小，直到接近最小可能误差值。

•泛化误差作为模型容量的函数呈u形曲线。

避免过拟合正则化:我们对学习算法所做的任何修改，旨在减少其泛化误差，而不是其训练误差。

常见的正则化形式

$$E_D(w) + \lambda E_W(w)$$

$$E_D(\boldsymbol{w}) = \sum_{i=1}^{m} \left(\boldsymbol{w^T x_i} - y_i\right)^2$$

第一个使依赖数据的误差，loss function 第二个是控制相对重要性的正则化系数，第三个是正则化项

一个简单的正则化（L2),避免过度拟合，L2广泛的用于过拟合上

$$E_W(w) = \frac{1}{2} w^T w$$

$$E_T(w) = \sum_{i=1}^{m} \left(\boldsymbol{w^T x_i} - y_i\right)^2 + \frac{\lambda}{2} w^T w \quad \textbf{Ridge regression}$$

Housing:

— Features: $x_1, x_2, \ldots, x_{100}$

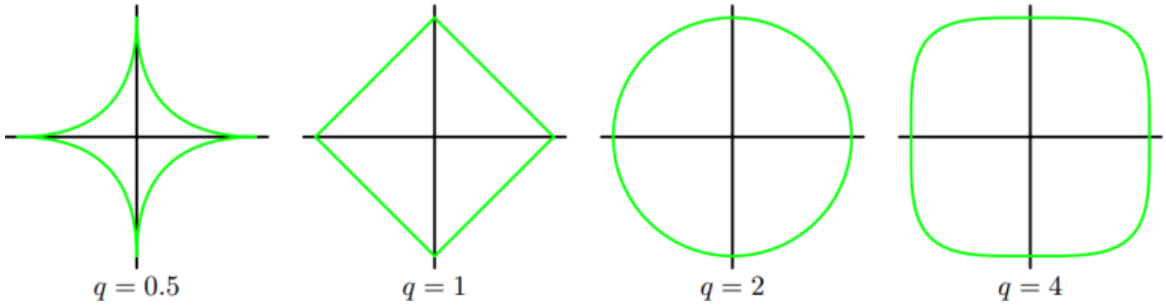
— Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

更一般的正则化器

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

$\boldsymbol{w} = [w_1, w_2]$



L1正则化

L1正则化→稀疏解。

它可以被认为类似于执行嵌入式特征选择，其中训练模型隐式地执行特征选择。

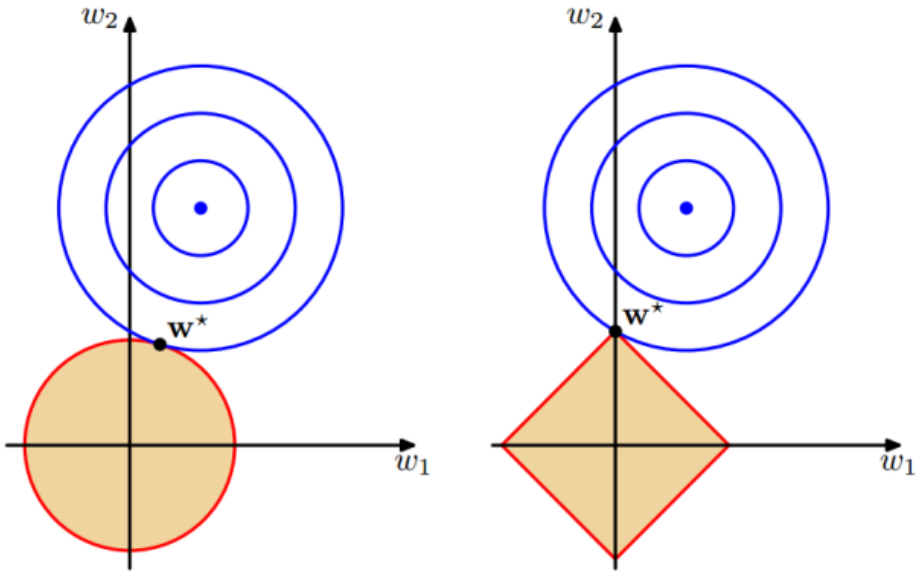
$$\min_{\boldsymbol{w}} \sum_{i=1}^{m}\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} - y_i\right)^2 + \lambda\|\boldsymbol{w}\|_1$$

L1 VS L2 Regularization

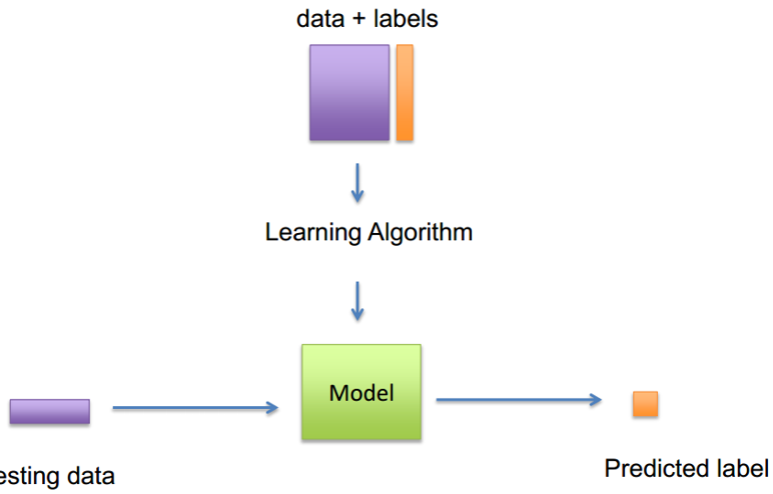
$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

w^* is the optimum value for w.

The lasso gives a sparse solution (w_1^*= 0).



监督学习（supervised learning)



数据+标签——>学习算法——>模型

训练数据——>模型——>预测标签

特征分类：

相关特征

不相关的特征

冗余特征（有用但是在其他特征出现过后就没有，可能是几个特征的组合）

特征选择：对于给定的一组特征集合X={X1,X2,X3...XD}和一个目标变量Y，‘

找到最小的集合S使得最大化分类的性能