[干货]基础机器学习算法

2016-10-08 晓惑 阿里技术

本篇内容主要是面向机器学习初学者,介绍常见的机器学习算法，当然,欢迎同行交流。

机器学习算法

什么是机器学习？

- 将无序数据转换为有用信息的方法

机器学习的价值是什么？

- 从数据中抽取规律，并用于解释数据或预知未来

举个栗子

- 收入预测

  算法1：收入 = a*年龄 + b*体重

  算法2：收入 = a*学历 + b*行业平均收入

哲学要回答的基本问题是从哪里来、我是谁、到哪里去，寻找答案的过程或许可以借鉴机器学习的套路：组织数据->挖掘知识->预测未来。组织数据即为设计特征，生成满足特定格式要求的样本，挖掘知识即建模，而预测未来就是对模型的应用。

机器学习算法

机器学习算法应用套路

- 数据处理（采集 + 去噪）
- 模型训练（特征 + 模型）
- 模型评估（MSE、F1Score、AUC + 调参）
- 模型应用（价值）

特征设计依赖于对业务场景的理解，可分为连续特征、离散特征和组合高阶特征。本篇重点是机器学习算法的介绍，可以分为监督学习和无监督学习两大类。

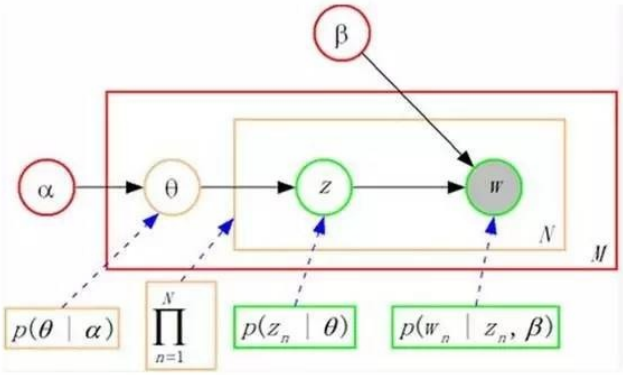
机器学习算法

◆ **无监督学习** – TopicModel

- TopicMode模型

  LSA =》 PLSA =》 LDA

- LDA算法

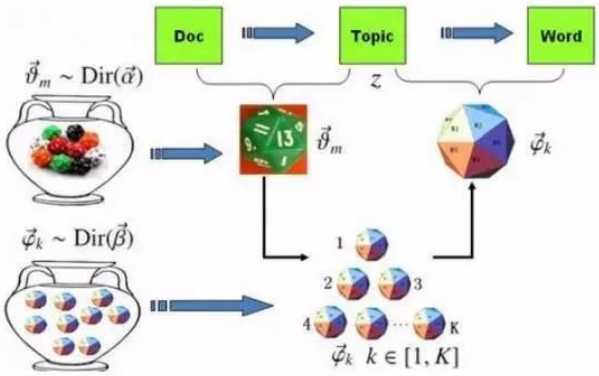


无监督学习算法很多，最近几年业界比较关注主题模型，LSA->PLSA->LDA为主题模型三个发展阶段的典型算法，它们主要是建模假设条件上存在差异。LSA假设文档只有一个主题，PLSA假设各个主题的概率分布不变（theta都是固定的），LDA假设每个文档和词的主题概率是可变的。

机器学习算法

◆ **无监督学习** – TopicModel

- LDA算法

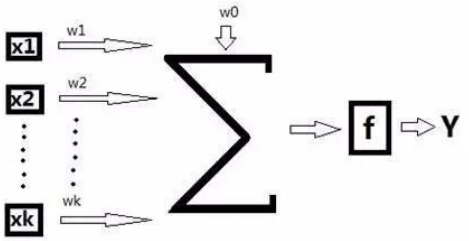


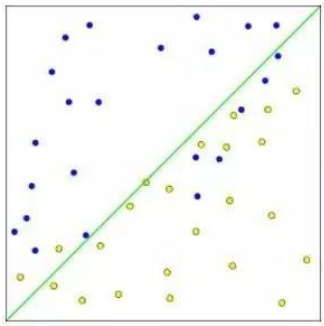
LDA算法本质可以借助上帝掷骰子帮助理解，详细内容可参加Rickjin写的《LDA数据八卦》文章，浅显易懂，顺便也科普了很多数学知识，非常推荐。

机器学习算法

◆ **监督学习** – 线性分类（感知器Perceptron）

- 模型算法





$$y = f\big(h(x)\big) = \begin{cases} 1, if\ h(x) > 0 \\ -1, if\ h(x) < 0 \end{cases}$$

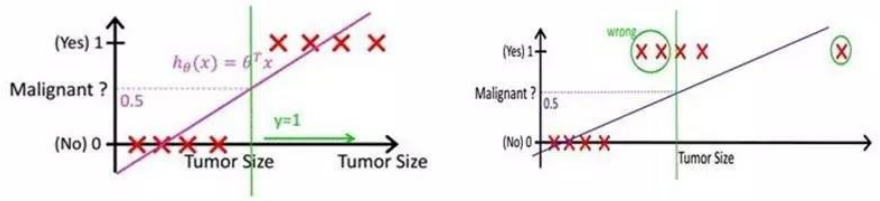
$$J(w) = \sum_{y \in Y} \delta_y w^T x$$

监督学习可分为分类和回归，感知器是最简单的线性分类器，现在实际应用比较少，但它是神经网络、深度学习的基本单元。

机器学习算法

◆ **监督学习** – 逻辑回归（分类）

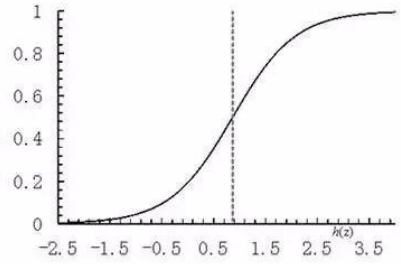
- 图形解释



$$y = \begin{cases} 1, & if\ h_\theta(\mathbf{x}) \geq 0.5 \ \Rightarrow\ \theta^T x \geq 0 \\ 0, & if\ h_\theta(\mathbf{x}) < 0.5 \ \Rightarrow\ \theta^T x < 0 \end{cases}$$

$$g(z) = \frac{1}{1+e^{-z}} \quad (1)$$

$$g(-z) = 1 - \frac{1}{1+e^{-z}} = \frac{1}{1+e^{z}} \quad (2)$$



线性函数拟合数据并基于阈值分类时，很容易受噪声样本的干扰，影响分类的准确性。逻辑回归（Logistic Regression）利用sigmoid函数将模型输出约束在0到1之间，能够有效弱化噪声数据的负面影响，被广泛应用于互联网广告点击率预估。

机器学习算法

◆ **监督学习** – 逻辑回归（分类）

- 模型算法

$$p(y=1|\theta) = h_\theta(\mathbf{x}) = \frac{1}{1+e^{-\theta^T\mathbf{x}}}$$

$$p(y=0|\theta) = 1 - h_\theta(\mathbf{x}) = \frac{e^{-\theta^T\mathbf{x}}}{1+e^{-\theta^T\mathbf{x}}}$$

$$p(y^{(i)}|\mathbf{x}^{(i)};\theta) = h_\theta(\mathbf{x}^{(i)})^{y^{(i)}}(1-h_\theta(\mathbf{x}^{(i)}))^{1-y^{(i)}}$$

$$\begin{cases} \text{When } y^{(i)} \text{ is 1, } p(y^{(i)}=1|\mathbf{x}^{(i)},\theta) = h_\theta(\mathbf{x}^{(i)}) \\ \text{When } y^{(i)} \text{ is 0, } p(y^{(i)}=0|\mathbf{x}^{(i)},\theta) = 1 - h_\theta(\mathbf{x}^{(i)}) \end{cases}$$

$$L(\theta) = p(\mathbf{y}|\mathbf{x};\theta) = \prod_{i=1}^{N} p(y^{(i)}|\mathbf{x}^{(i)},\theta)$$

$$= \prod_{i=1}^{N} h_\theta(\mathbf{x}^{(i)})^{y^{(i)}}(1-h_\theta(\mathbf{x}^{(i)}))^{1-y^{(i)}}$$

机器学习算法

◆ **监督学习** – 逻辑回归（分类）

- 模型算法

$$l(\theta) = logL(\theta)$$

$$= \prod_{i=1}^{N} h_\theta(\mathbf{x}^{(i)})^{y^{(i)}}(1-h_\theta(\mathbf{x}^{(i)})^{(1-y^{(i)})}$$

$$= \sum_{i=1}^{N} y^{(i)} log(h_\theta(x^{(i)})) + (1-y^{(i)})(1-log(h_\theta(x^{(i)}))$$

$$\frac{\partial l(\theta)}{\partial \theta_j} = (y^i - h_\theta(x^{(i)}))x_j$$

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

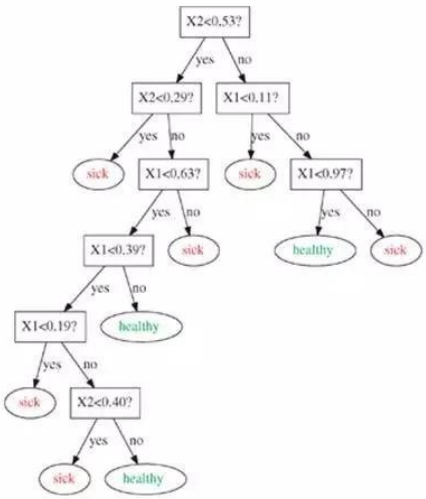
$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1-h_\theta(x)) & \text{if } y = 0 \end{cases}$$

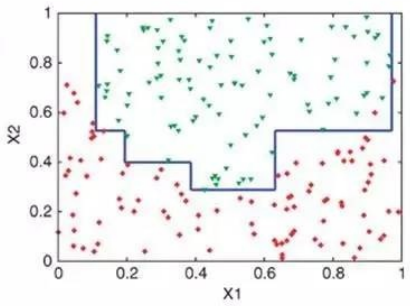
逻辑回归模型参数可以通过最大似然求解，首先定义目标函数L(theta)，然后log处理将目标函数的乘法逻辑转化为求和逻辑（最大化似然概率 -> 最小化损失函数），最后采用梯度下降求解。

机器学习算法

◆ **监督学习** – 决策树（分类）

- 模型算法（ID3、C4.5）





机器学习算法

◆ **监督学习** – 决策树（分类）

- 模型算法（ID3、C4.5）

- Main loop:
  - 1. Select a feature F best classifies examples
  - 2. Create a node using F, separate the data set with different values of F
  - 3. Recursively build sub-tree
  - 4. Stop until the samples have the same category: build a leaf node with this category

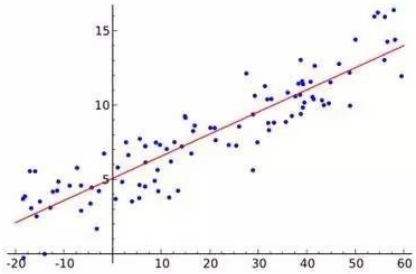
- 不同算法定义的增益函数不同

相比于线性分类去，决策树等非线性分类器具有更强的分类能力，ID3和C4.5是典型的决策树算法，建模流程基本相似，两者主要在增益函数（目标函数）的定义不同。

机器学习算法

◆ **监督学习** – 线性回归

- 模型算法



$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)} \qquad \varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$$

$$L(\theta) = p(\overrightarrow{y}|X;\theta) = \prod_i^m p(y^{(i)}|x^{(i)};\theta) = \prod_i^m \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}}$$

$$maxL(\theta) \Rightarrow min\frac{1}{2}\sum_i^m (y^{(i)} - \theta^T x^{(i)})^2$$

线性回归和线性分类在表达形式上是类似的，本质区别是分类的目标函数是离散值，而回归的目标函数是连续值。目标函数的不同导致回归通常基于最小二乘定义目标函数，当然，在观测误差满足高斯分布的假设情况下，最小二乘和最大似然可以等价。

机器学习算法

◆ **监督学习** – 线性回归

- 训练效率

Batch gradient descent

- use all the data in one iteration

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{N} (y^{(i)} - f_\theta(x^{(i)}) x_j^{(i)}$$

}

- Can convergent to global optima
- Slowly for each iteration, especially when N is large

Stochastic gradient descent

- use one sample per iteration

Repeat until convergence {

for $i = 1 \ldots N$ {

$$\theta_j := \theta_j + \alpha (y^{(i)} - f_\theta(x^{(i)}) x_j^{(i)}$$

}

}

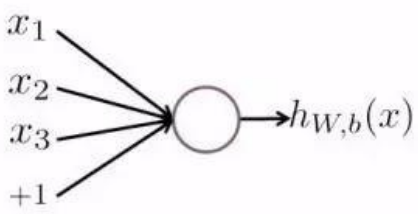
- Convergent faster
- May not reach the global optima

当梯度下降求解模型参数时，可以采用Batch模式或者Stochastic模式，通常而言，Batch模式准确性更高，Stochastic模式复杂度更低。

机器学习算法

◆ **Deep Learning**（Neural Network）

- Neuron

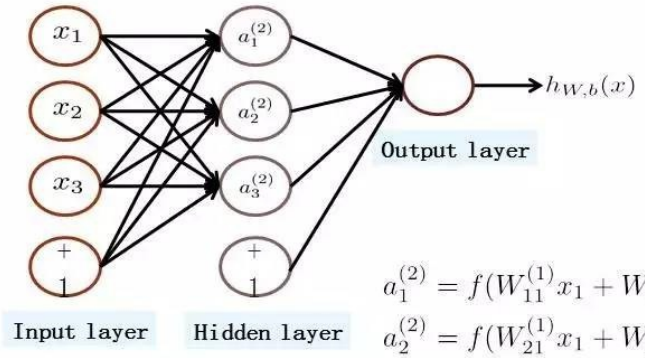


$$h_{w,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$$

机器学习算法

◆ **Deep Learning**（Neural Network）

- Neural Network



$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$
$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$
$$a_3^{(3)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)})$$

$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}x_1 + W_{12}^{(2)}x_2 + W_{13}^{(2)}x_3 + b_1^{(2)})$$

上文已经提到，感知器虽然是最简单的线性分类器，但是可以视为深度学习的基本单元，模型参数可以由自动编码（Auto Encoder）等方法求解。

机器学习算法

◆ **Deep Learning**（Neural Network）

- **核心思想**

  把learning hierarchy 看做一个network，则

  - 无监督学习用于每一层网络的pre-train(e.g. autoEncoder)
  - 每次用无监督学习只训练一层，将其训练结果作为其higher一层的输入
  - 用监督学习调整所有层

- **DL在图像、语音等领域取得了很优异的效果**

深度学习的优势之一可以理解为特征抽象，从底层特征学习获得高阶特征，描述更为复杂的信息结构。例如，从像素层特征学习抽象出描述纹理结构的边缘轮廓特征，更进一步学习获得表征物体局部的更高阶特征。

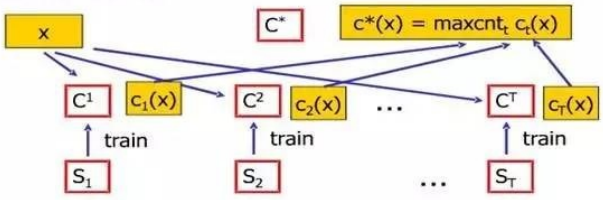
机器学习算法

◆ **Model Ensemble**

- 设计初衷：三个臭皮匠赛过诸葛亮
- Bagging
  - 原则：模型之间应有差异性（e.g. 数据、特征、参数）
  - 方法：regression Averaging / classification voting
  - 调优：基于validation data调参
  - 举例：

    regression：a*人气模型 + b*LD模型 + c*LC模型 + ...

    classification：



俗话说三个臭皮匠赛过诸葛亮，无论是线性分类还是深度学习，都是单个模型算法单打独斗，有没有一种集百家之长的方法，将模型处理数据的精度更进一步提升呢？当然，Model Ensembel就是解决这个问题。Bagging为方法之一，对于给定数据处理任务，采用不同模型/参数/特征训练多组模型参数，最后采用投票或者加权平均的方式输出最终结果。

机器学习算法

◆ Model Ensemble

- ➢ 设计初衷：三个臭皮匠赛过诸葛亮
- ➢ Boosting
  - 原则：模型之间应有差异性（e.g. 数据）
  - 方法：每次迭代，错误样本增加权重
  - 调优：基于validation data调参
  - 举例：AdaBoost

Input: (x_i, y_i)

Init: $\alpha_1\ (i) = \frac{1}{n}$

Training: for $t = 1,2, \ldots, m$

弱分类器h_t, 计算$e_t = \sum w_t(i)\,[h_t(x_i) \neq y_i]$, $\alpha_t = \frac{1}{2}\ln\left(\frac{1-e_t}{e_t}\right)$

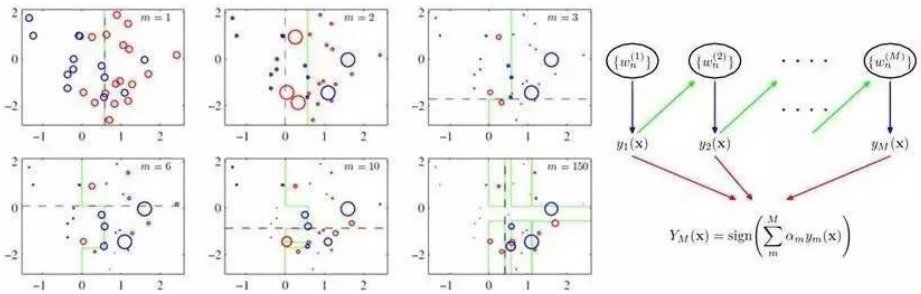
if $h_t(x_i) \neq y_i$, $w_{t+1}(i) = w_t(i) \cdot e^{\alpha_t}/Z_t$, else $w_{t+1}(i) = w_t(i) \cdot e^{-\alpha_t}/Z_t$

Output: $H(x) = sign(\sum \alpha_t \cdot h_t(x))$

机器学习算法

◆ Model Ensemble

- ➢ 设计初衷：三个臭皮匠赛过诸葛亮
- ➢ Boosting
  - 举例：AdaBoost



Boosting为Model Ensemble的另外一种方法，其思想为模型每次迭代时通过调整错误样本的损失权重提升对数据样本整体的处理精度，典型算法包括AdaBoost、GBDT等。

机器学习算法

◆ **Model Ensemble**

- 设计初衷：三个臭皮匠赛过诸葛亮
- Bagging与Boosting

	Training	Testing
Bagging	训练集独立	可并行
Boosting	训练集依赖	需串行，准确性高，易过拟合

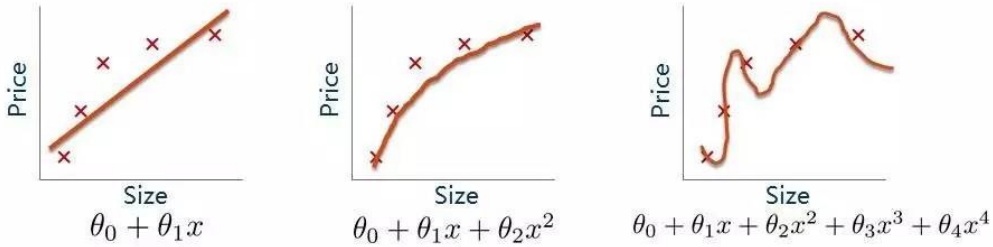
- Dropout（DL）

不同的数据任务场景，可以选择不同的Model Ensemble方法，对于深度学习，可以对隐层节点采用DropOut的方法实现类似的效果。

机器学习算法

◆ **模型调参**

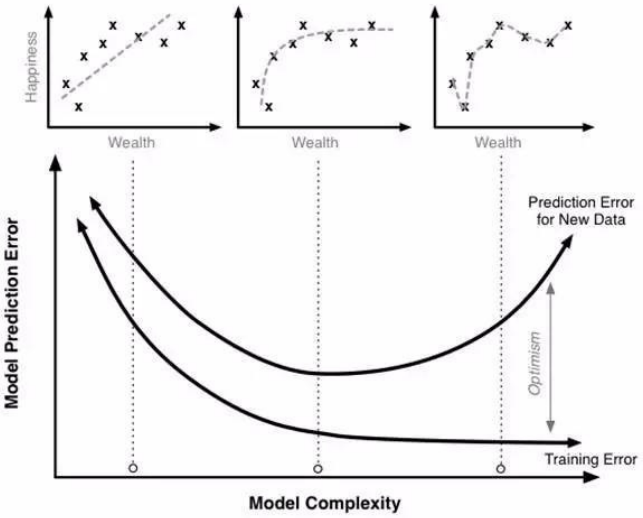
- 欠拟合与过拟合



机器学习算法

◆ **模型调参**

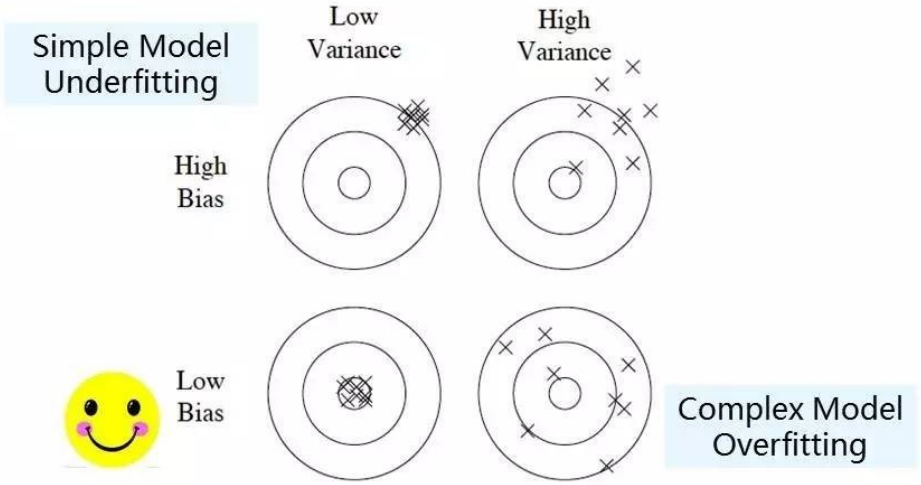
- 欠拟合与过拟合



机器学习算法

◆ **模型调参**

- 偏置（Bias）与方差（Variance）

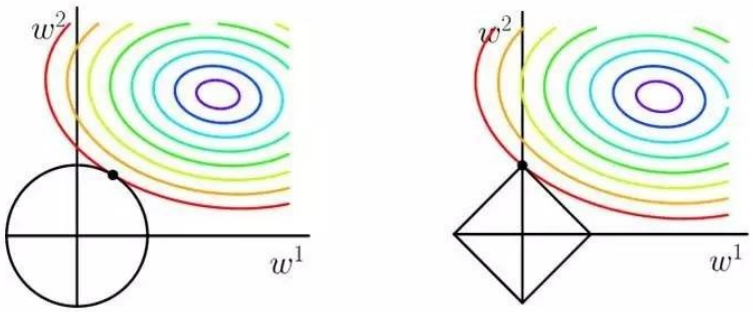


介绍了这么多机器学习基础算法，说一说评价模型优劣的基本准则。欠拟合和过拟合是经常出现的两种情况，简单的判定方法是比较训练误差和测试误差的关系，当欠拟合时，可以设计更多特征来提升模型训练精度，当过拟合时，可以优化特征量降低模型复杂度来提升模型测试精度。

机器学习算法

◆ **模型调参**

- L2正则与L1正则

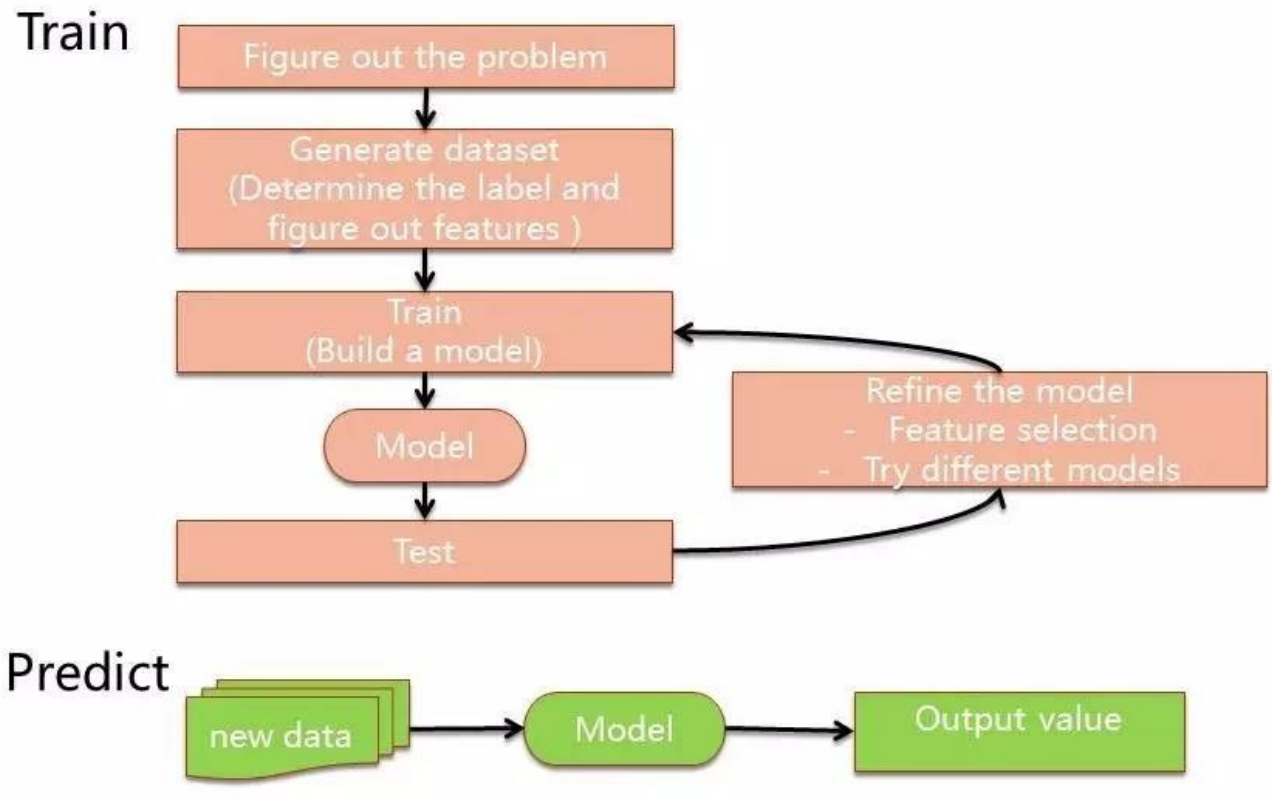


$$J(\theta) = Cost(E) + Cost(W)$$

特征量是模型复杂度的直观反映，模型训练之前设定输入的特征量是一种方法，另外一种比较常用的方法是在模型训练过程中，将特征参数的正则约束项引入目标函数/损失函数，基于训练过程筛选优质特征。

机器学习算法

- **模型解决问题流程**



模型调优是一个细致活，最终还是需要能够对实际场景给出可靠的预测结果，解决实际问题。期待学以致用！

关注阿里巴巴官方技术号：