C1-10

Machine Learning
by Andrew Ng, Stanford Engineering

Xiaojie Zhou

szxjzhou@163.com

2016.9.13

第十一集：贝叶斯统计正则化

- 贝叶斯统计正则化(Bayesian Statistics Regularization)
- 在线学习(Online Learning)
- 对使用机器学习算法的一些建议(Advice of Applying Machine Learning Algorithms)

第十一集：贝叶斯统计正则化

- 贝叶斯统计正则化
  - 之前已经讲到过通过进行交叉验证和特征选择以选择合适的模型，避免欠拟合和过拟合的问题的出现
  - 之前讲到的避免过拟合的方法主要集中在模型选择和特征选择上，实际上除此以外还可以通过控制参数θ来避免过拟合，在开始这部分内容之前先看一下对于参数θ的两种观点
    - 在频率学派(frequentist)看来参数θ是未知的但是确定的值(constant-valued but unknown) .由于它是确定的，因此我们可以通过统计过程（比如极大似然）来预测它的值。从而得到了如下的参数θ计算方法

$$\theta_{\mathrm{ML}} = \arg\max_{\theta} \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta)$$

    - 而在贝叶斯学派(Bayesian)看来参数θ是未知的与此同时还是随机变量，因此不能找出某一个确定的值，只能指定参数θ所服从的先验分布(prior distribution) p(θ)
      - 这个先验分布描述的是在最原始的情况下我们所认为的θ应该是个什么样子的，而在一般情况下可将p(θ)指定为均值为0，协方差矩阵为$\tau^2 I$的高斯分布（τ为一个参数）

第十一集：贝叶斯统计正则化

- 贝叶斯统计正则化
  - 因此在贝叶斯学派下需要基于参数θ所服从的先验分布p(θ)，测试集S的信息来得到参数θ在测试集S下的后验概率

$$
\begin{aligned}
p(\theta|S) &= \frac{p(S|\theta)p(\theta)}{p(S)} \\
&= \frac{\left(\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta)\right)p(\theta)}{\int_{\theta}\left(\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta)p(\theta)\right)d\theta}
\end{aligned}
$$

在这里假设了测试集S中一共有m个样本，而这些样本间相互独立。其中p(y|x,θ)类似于之前的p(y|x;θ)，这里写成这样只是因为θ将作为一个随机变量参与到其中，而不是一个确定值。举个Logistic回归的例子，其p(y|x,θ)定义为：

$$
\begin{aligned}
p(y^{(i)}|x^{(i)},\theta) &= h_\theta(x^{(i)})^{y^{(i)}}(1-h_\theta(x^{(i)}))^{(1-y^{(i)})} \\
h_\theta(x^{(i)}) &= 1/(1+\exp(-\theta^T x^{(i)}))
\end{aligned}
$$

  - 通过上面这些信息即可通过下面的公式对于新的测试数据x进行类标预测，也可以对于期望进行计算

$$
p(y|x,S) = \int_{\theta} p(y|x,\theta)p(\theta|S)d\theta \qquad \mathrm{E}[y|x,S] = \int_{y} yp(y|x,S)dy
$$

第十一集：贝叶斯统计正则化

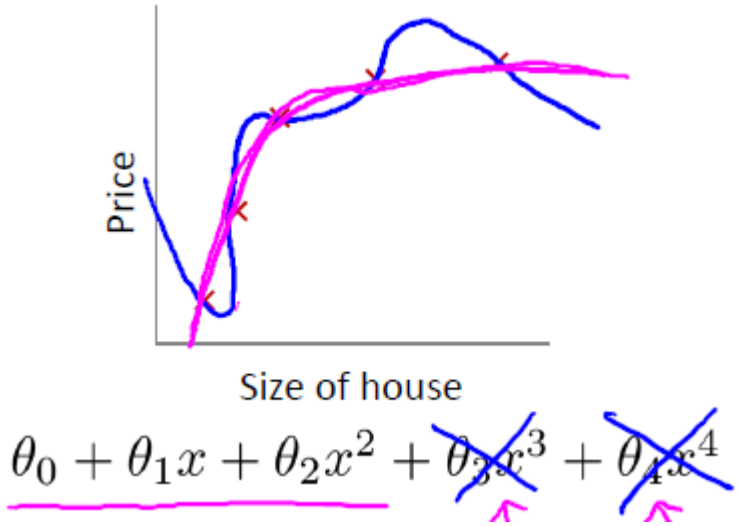
- 贝叶斯统计正则化
  - 这种贝叶斯统计方法看上去比较客观，但实际上参数θ在测试集S下的后验概率极其难算
    - 主要在于后验概率的分子部分的关于参数θ的积分，因为这里面的θ很可能是高维的，而且我们很难找到对应的原函数
  - 因此只能对参数θ进行近似的计算，这被称为参数θ的最大后验估计 (Maximum A Posteriori (MAP) estimate)

$$\theta_{\mathrm{MAP}} = \arg\max_{\theta} \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}, \theta)p(\theta)$$

对比原来的极大似然公式多出了p(θ)的部分，一般会将p(θ)指定为均值为0，协方差矩阵为$\tau^2 I$（τ为一个参数）的高斯分布。选取这一分布的原因是，这个高斯分布的均值为0，因此可以认为数据散布在0的附近（散布的程度由系数τ所控制），从而在这种情况下最后得到的θ的范数比用之前的极大似然公式计算出的θ的范数小（因为θ中有相当部分的值都近似为0），从而导致了样本x中对应的这些特征几乎不起作用。这种方法给了进行特征选择一种可替代交叉验证的方案，被广泛用于文本分类这种θ维数很大的问题中

第十一集：贝叶斯统计正则化

- 贝叶斯统计正则化
  - 这种方法被称为贝叶斯统计正则化方法，下图说明了贝叶斯统计正则化比起原来极大似然方法的区别



对于这些样本点来说，如果我们采用了五阶的曲线对于训练集中样本用极大似然的方法进行拟合，会得到如图蓝线所示的过拟合曲线。而贝叶斯统计正则化加上p(θ)后（假设p(θ)服从均值为0，协方差矩阵为$\tau^2 I$（τ为一个参数）的高斯分布）随着τ的缩小会使得θ中有相当部分的值都近似为0，从而间接起到了降低维数的效果，在图像上会使得曲线的波动更不明显，从而减少了过拟合的可能性

第十一集：贝叶斯统计正则化

- 在线学习
  - 目前所涉及的机器学习方法都可以统称为批学习算法(batch learning)
    - 批学习算法的特征为先用训练集对模型进行统一的训练，然后再用测试集对训练好的模型进行测试
    - 而与此同时还有一种学习方法，需要在学习的过程中反复进行测试，而不是在训练好模型后再进行集中测试，这种方法被称为在线学习算法(online learning)
      - 在在线学习算法中，对于一个分类的问题，首先会在不告知任何信息的情况下先对第一个样本的类标进行预测，然后告知第一个样本的真实类标并用此调整模型。之后再尝试对第二个样本的类标进行预测（注意此时不再对于第一个样本的类标进行预测），然后告知第二个样本的真实类标并用此调整模型。如此一直持续到最后一个样本
      - 这种方法常被用于样本不能被同时取得的情况，使得学习的过程可持续扩展新的样本，而不需等到所有样本都取得后再进行预测
      - 而在线学习的目标是最小化总在线学习误差，其衡量方法为预测类标错误的样本数

第十一集：贝叶斯统计正则化

- 在线学习
  - 下面以感知机算法为例来看之前讲过的机器学习方法是如何应用在在线学习上的
    - 首先先看一下在之前的批学习算法中感知机算法是如何进行学习的
      - 对于感知计算法来说可以认为是Logistic回归的极端情况，其预测函数为

$$h_\theta(x) = g(\theta^T x) \qquad g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0. \end{cases}$$

      - 作为Logistic回归的极端情况，感知机算法同样可以采用Logistic回归中参数θ的更新方法进行更新

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

$$\theta := \theta + yx.$$

上面这个式子显示的是之前讲到的批学习算法中的Logistic回归中参数θ的更新方法，显然这个式子也可被用在在线学习方法中。在这个式子中原来类标y在{0,1}之间取值，最后的效果为如果当前预测正确，那么参数θ不发生改变（因为y-h(x)=0）；如果当前预测错误，那么参数θ=θ±2αx。现在为了简化模型令类标y在{-1,1}之间取值，并将α的值设为1/2（因为α的值只影响参数θ变化的程度，并不影响最终的分类结果），从而得到了下面的这个式子对于当前预测错误的情况进行参数θ的调节（对于预测正确的样本不进行参数θ的调节），而这样更改的好处在于可以用核方法计算yx

第十一集：贝叶斯统计正则化

- 在线学习
  - 接下来进一步讨论感知机算法在在线学习中的一个很好的性质
    - 定理(Block, 1962, and Novikoff, 1962)：给定m个样本组成的训练集，并假定训练集中的输入特征的维数和值均有限（即$||x||\leq D$），并存在一个单位向量u使得每组训练样本满足$y(u^Tx)\geq\gamma$（换句话说就是一方面要保证训练集可被分隔开，另一方面要保证距离分隔线的距离至少为γ），感知机算法在在线学习中的总在线学习误差不超过$(D/\gamma)^2$
      - 这个结论的意义在于在满足上述条件的情况下感知机算法在在线学习中的总在线学习误差的最坏情况与样本个数和输入特征的维数无关，因此用很少的样本同样可以训练出一个不错的模型，也不需要进行特征选择

由于感知机算法在遇到错误时才进行更新，因此不妨假设当前遇到第 k 个错误样本（这组样本不一定是第 k 组，因此假设这组样本是训练集中的第 i 组，而这里的 k 可以看成是对于训练集中前 i 组样本组成的小训练集的总在线学习误差）。由于前 i-1 组样本已经完成了训练，我们不妨假设训练的参数θ的结果为θk，因此有：

$$(x^{(i)})^T\theta^{(k)}y^{(i)}\leq 0$$

由于这组样本遇到了错误，因此我们需要对θk 用公式θ=θ+yx 更新到θk+1

$$\begin{aligned}(\theta^{(k+1)})^Tu &= (\theta^{(k)})^Tu+y^{(i)}(x^{(i)})^Tu\\ &\geq (\theta^{(k)})^Tu+\gamma\end{aligned}$$

而在这里θk^Tu 又可以继续展开为：$(\theta k)^Tu\geqslant(\theta k\text{-}1)^Tu+\gamma$，如此一直展开下去易知（可以不失一般性地假设初始化的θ=0）：

$$(\theta^{(k+1)})^Tu\geq k\gamma$$

而与此同时根据$||x||\leqslant D$ 又有

$$\begin{aligned}||\theta^{(k+1)}||^2 &= ||\theta^{(k)}+y^{(i)}x^{(i)}||^2\\ &= ||\theta^{(k)}||^2+||x^{(i)}||^2+2y^{(i)}(x^{(i)})^T\theta^{(i)}\\ &\leq ||\theta^{(k)}||^2+||x^{(i)}||^2\\ &\leq ||\theta^{(k)}||^2+D^2\end{aligned}$$

同样的道理这里$||\theta k||^2$ 又可以继续展开为：$||\theta k||^2\leqslant||\theta k\text{-}1||^2+D^2$，，如此一直展开下去易知（初始化的θ=0）：

$$||\theta^{(k+1)}||^2\leq kD^2$$

联立上述关于θk+1 的两条不等式结论可得（在这里面利用了$||\theta k+1||=||\theta k+1||\cdot||u||\geqslant(\theta k+1)^Tu$）

$$\begin{aligned}\sqrt{k}D &\geq ||\theta^{(k+1)}||\\ &\geq (\theta^{(k+1)})^Tu\\ &\geq k\gamma.\end{aligned}$$

通过这个式子可以发现，对于任意一个训练集来说 k 的值最多为$(D/\gamma)^2$，即最大总在线学习误差不超过$(D/\gamma)^2$

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 下面的内容基于Andrew Ng.在实践过程中对于运用机器学习方法的一些建议，与理论关系不大，基于Andrew Ng.的个人实践经验
  - 内容从下面要点展开
    - 1、如何在实践上改进机器学习算法的效果
    - 2、误差分析
    - 3、如何着手解决一个机器学习问题
      - 特别是如何分析出该问题的瓶颈以避免过早优化（过早优化指的是我们对于并不关键的代码做了优化）

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 在实践上改进机器学习算法的效果
    - 比如说当前有一个垃圾邮件分类的问题，采用了如下的带正则化项的Logistic分类的办法，但是得到了一个很大的误差，应该如何处理？

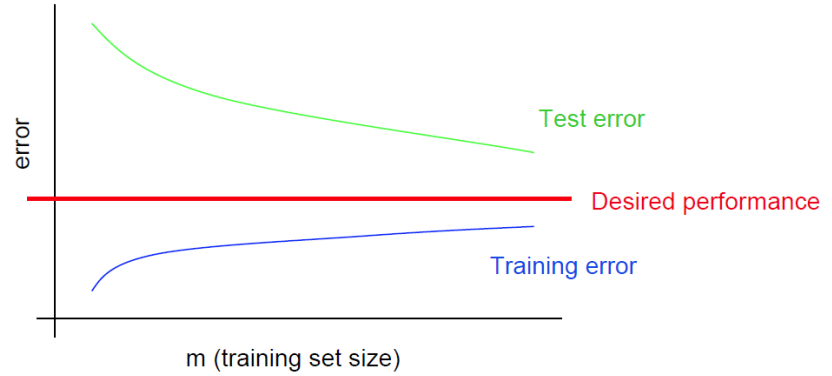
$$\max_{\theta} \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)},\theta) - \lambda||\theta||^2$$

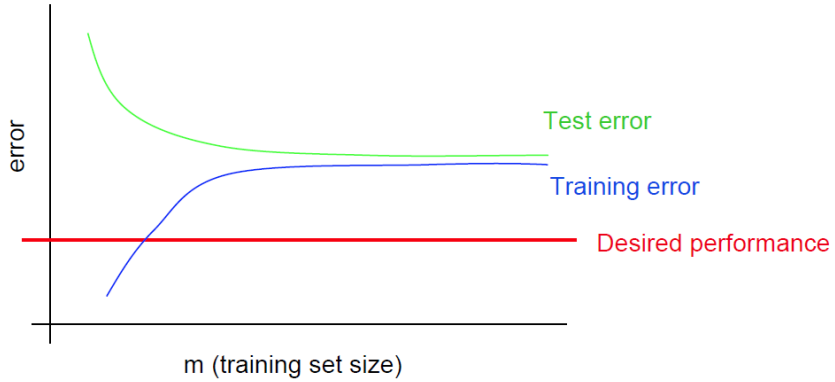
在这里正则化项的意义在于控制了θ值的模，从而使得样本x中对应的这些特征几乎不起作用，在图像上使得曲线更加平滑，降低了过拟合的可能性

    - 这时候盲目采取应对措施费时费力还效果不佳，其实可以从下面这些问题入手
      - 1、是否出现了欠拟合和过拟合的问题？
      - 2、是否设计了不恰当的目标函数？
      - 3、是否采用了不恰当的最优化方法？
      - …

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 在实践上改进机器学习算法的效果
    - 1、欠拟合和过拟合





左边两幅图像很好地显示了过拟合（高方差）和欠拟合（高偏差）的特征

对于过拟合问题来说，随着训练集的增大，测试误差会持续下降，而且训练误差和测试误差的差值比较大。那在这情况下很可能样本太少从而不足以反映全局情况，也很有可能是因为特征太多，拟合了不需要的特征。因此对此的解决办法是：增大训练集、进行特征选择以减少特征

对于欠拟合问题来说，训练误差本身就比较大，但训练误差和测试误差的差值比较小。那在这情况下很可能忽略了某些重要特征。因此对此的解决办法是：进行特征选择以增加特征或更换特征

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 在实践上改进机器学习算法的效果
    - 2、目标函数的设计
      - 对于机器学习问题来说往往有一些参数需要人工根据经验进行设置，比如说正则化项中的λ，SVM软间隔参数C，而这些参数设置错误会给最终的效果带来不利影响。与此同时也有可能我们选用了错误的模型，从而得到了不适合此问题的目标函数
      - 对此的解决方法：尝试换不同的目标函数和参数
    - 3、最优化方法
      - 实际上我们可以采用的最优化方法有很多（之前就已讲过梯度下降法、牛顿方法、顺序最小优化方法），有可能我们选取了一种不恰当的最优化方法，或者我们的收敛条件过松导致了结束时得到了一个并不准确的值
      - 对此的解决方法：尝试换一种最优化方法或者收紧原算法的收敛条件
    - 但怎样发现是目标函数的问题还是最优化方法的问题？

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 在实践上改进机器学习算法的效果
    - 要想发现是目标函数的问题还是最优化方法的问题需要借助于一个能够得到更优参数θ值的算法
      - 比如说在垃圾邮件问题中，考虑到特征过多需要进行特征选择和算法运行速度的问题，决定采用带正则化项的Logistic回归方法。当发现带正则化项的Logistic回归方法对于参数θ值的学习效果不佳时（测试误差大），可以采用SVM等其他方法进行参数θ值的学习
        - 当我们发现某种算法学习得到的参数θ的测试效果比原方法更优时，我们可以将该θ带入到原方法的目标函数中
          - 比如在这里我们发现SVM学习得到的θ(SVM)值的测试效果比带正则化项的Logistic回归学习得到的θ(BLR)值测试效果好，那么我们就可以将θ(SVM)带入到带正则化项的Logistic回归的目标函数中，观察在目标函数中哪个值更优

$$J(\theta) = \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)}, \theta) - \lambda||\theta||^2 \qquad J(\theta_{\text{SVM}}) > J(\theta_{\text{BLR}})?$$

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 在实践上改进机器学习算法的效果
    - 这时候无非以下两种情况

$$J(\theta_{\mathrm{SVM}}) > J(\theta_{\mathrm{BLR}})$$

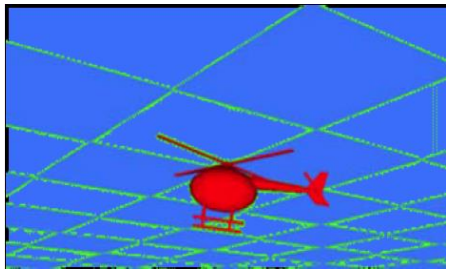
在这种情况下SVM学习得到的θ(SVM)值在目标函数中的表现比带正则化项的Logistic回归学习得到的θ(BLR)值好（因为我们需要极大化J(θ)），这就说明在优化目标函数的过程中出现了问题，没有优化到一个更优的值。对此可以尝试换一种最优化方法或者收紧原算法的收敛条件

$$J(\theta_{\mathrm{SVM}}) \leq J(\theta_{\mathrm{BLR}})$$

在这种情况下SVM学习得到的θ(SVM)值在目标函数中的表现比带正则化项的Logistic回归学习得到的θ(BLR)值差（因为我们需要极大化J(θ)），这就说明在目标函数本身出现了问题，不能真实反映我们的优化目标。对此可以尝试换不同的目标函数和参数

第十一集：贝叶斯统计正则化

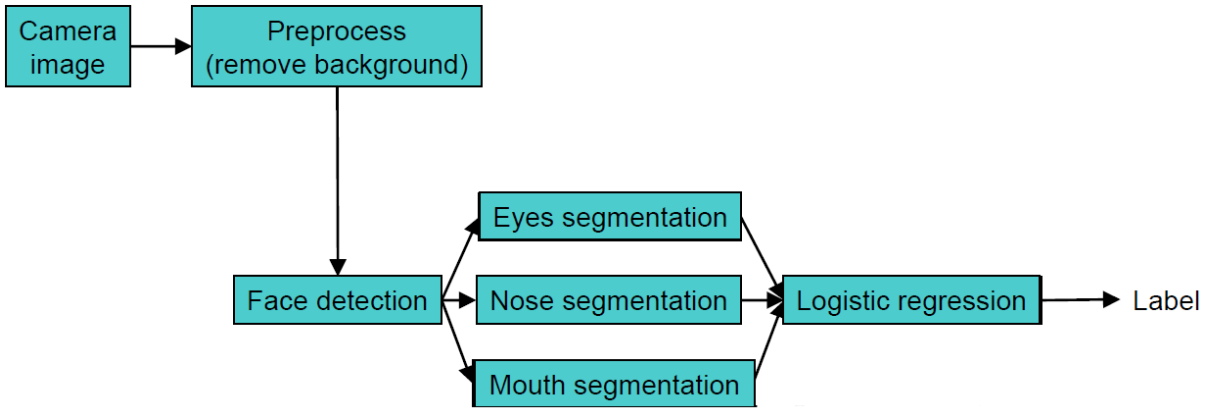
- 对使用机器学习算法的一些建议
  - 在实践上改进机器学习算法的效果
    - 下面看一个更加实际的例子，对于自动驾驶飞行器控制器的设计



      - 对于自动驾驶飞行器控制器的设计过程主要分三个部分
        - 1、构建自动驾驶飞行器控制器（地面模拟器）
        - 2、设置一个目标函数（比如说：模拟位置和真实位置的距离）
        - 3、采用算法最小化目标函数，得到最优参数
      - 如果最终的运行效果不佳，那么我们就可以怀疑
        - 1、模拟器出了问题：模拟器上运行的效果很好但实际飞行效果差
        - 2、目标函数设置错误：将飞机调整到手动飞行模式，此时会得到一组非常好的参数θ(MAN)，将其代入目标函数后发现$J(\theta(MAN)) \geq J(\theta(LEARN))$
        - 3、优化算法有问题：将飞机调整到手动飞行模式，此时会得到一组非常好的参数θ(MAN)，将其代入目标函数后发现$J(\theta(MAN)) < J(\theta(LEARN))$

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 误差分析
    - 对于一个机器学习实际应用来说可能由多个组成部分，而每个部分都有对应的误差可以进行优化
      - 因此在实际应用过程中我们不仅要对整个应用进行误差测试，更重要的是要对每个部分进行测试，从而找到问题所在
      - 而对此一种分析方法是按顺序依次对各个模块用人工替换机器学习的方法进行考察



Component	Accuracy
Overall system	85%
Preprocess (remove background)	85.1%
Face detection	91%
Eyes segmentation	95%
Nose segmentation	96%
Mouth segmentation	97%
Logistic regression	100%

对于这个例子，完全用机器学习的方法最终测试的准确率为85%，此时我们用人工预处理替代机器预处理再进行测试发现准确率上升了0.1%，变成85.1%.接下来我们用人工人脸识别替代机器人脸识别再进行测试发现准确率上升了5.9%，变成91%.由此一直下去，从中就可以发现问题主要出现在哪些模块上

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 误差分析
    - 上面说的那种方法试图对各个模块用人工替换机器学习的方法进行考察
    - 实际上还有一种相反的方法
      - 假设直接用Logistic回归的方法进行垃圾邮件的分类效果不太佳，这时考虑在里面加入一些功能使得垃圾邮件的分类效果由94%提升至99.9%
      - 但里面有些功能有用，有些没有。为检验每个模块对性能提升的程度，我们可以依次撤销一些模块

- Spelling correction.
- Sender host features.
- Email header features.
- Email text parser features.
- Javascript parser.
- Features from embedded images.

Component	Accuracy
Overall system	99.9%
Spelling correction	99.0
Sender host features	98.9%
Email header features	98.9%
Email text parser features	95%
Javascript parser	94.5%
Features from images	94.0%

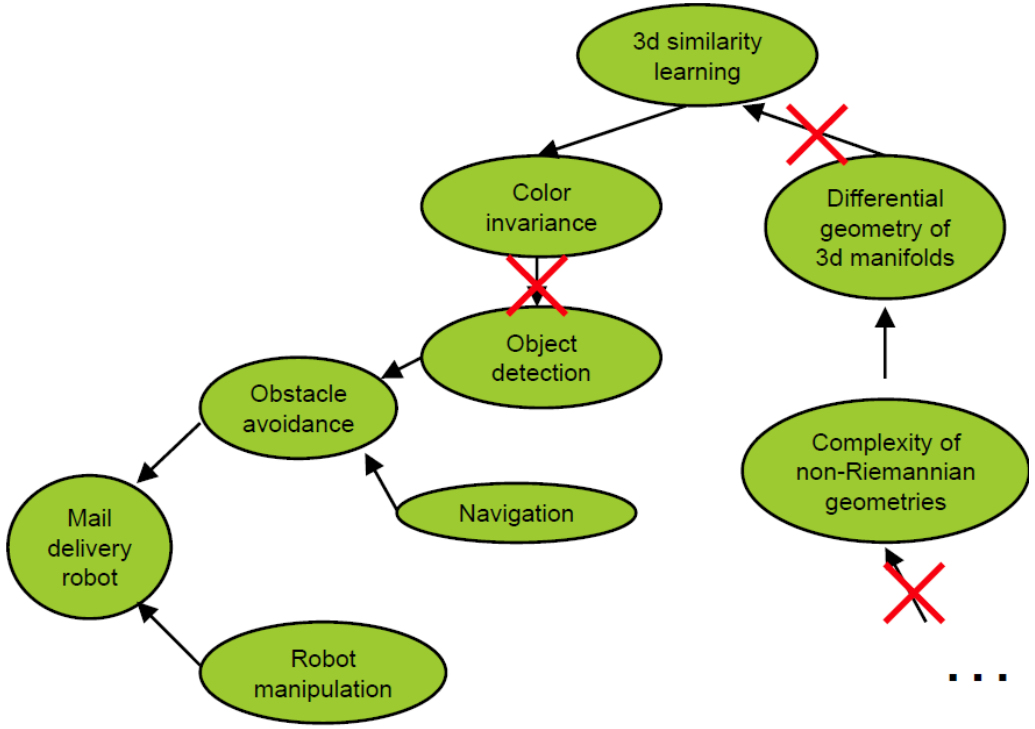
对于这个例子，完全用机器学习的方法最终测试的准确率为99.9%，此时我们撤销掉拼写纠正准确率降低了0.9%，变为了99%；再撤销掉发送者主站的特征准确率降低了0.1%，变为了98.9%...从中可以发现提升性能关键在哪些模块上

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 着手解决一个机器学习问题
    - 一般来说存在下面两种设计的思路
      - 1、Careful design
        - 对于流程结构、数据集、所用算法等都先进行仔细的考量
        - 实现这一算法
      - 评价：可以得到一个理论上非常优雅的问题解决方案，但流程较长
      - 2、Build-and-fix
        - 先着手实现一些基本的功能
        - 边测试边调整之前的算法
      - 评价：很容易出现各种各样的问题，不过实现速度快，能及早将产品投入市场
    - 对于一个市场化的软件来说，不要在一开始试图花很多时间对于某个功能模块进行优化，先保证基本功能的运行
      - 因为你在没有得到完整系统并进行误差分析前是不知道哪些模块是关键的

第十一集：贝叶斯统计正则化

- 对使用机器学习算法的一些建议
  - 对此还有一点需要注意的是不要过早深究或优化某些细节的问题



这里面给出了一个反例，比如在考虑设计一个邮件投递机器人的时候，将问题分成了避障和机器人动作两个模块。而在考虑避障的时候又考虑障碍物的识别和导航，接下来继续从障碍物的识别中又引出了颜色的识别，然后从颜色的识别中又引出了后续的很多相关问题。这样做将会导致花了很多时间在研究这些细枝末节，而这些细枝末节有可能对于整体性能的作用微乎其微