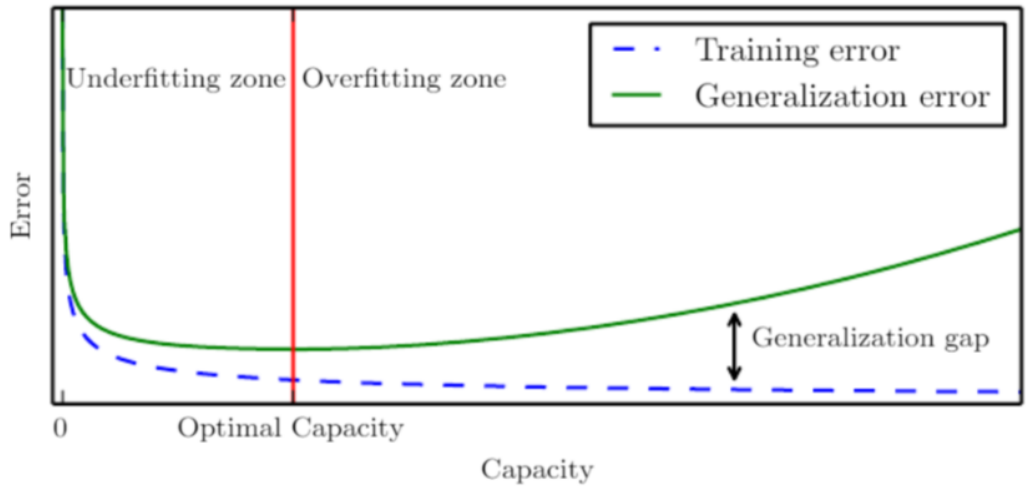
2.5 Machine Learning Theory

Introduction

在 2.1 当中，我们得知

1. 训练误差 training error 常随着模型容量 capacity 的增大而减小；
2. 泛化误差 generalization error 随着模型容量的增大先减小后增大。



这一节的目的在于：

1. 在回归当中使用偏差 bias 和方差 variance 的概念解释诸多现象；
2. 在分类当中引入模型容量的测量方法 VC dimension，以及 generalization gap 是怎样受到 VC dimension 以及样本大小 sample size 的影响的。

注意本节的标注符号和前文有所不同，以保持和相关文献的一致性。

回归问题：偏差和方差的 trade-off

问题描述：给定训练集$S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$，其中$y_i \in \mathbb{R}$，其对应的回归方程所组成的假设空间 hypothesis class $\mathcal{H}$：

$$\mathcal{H} = \{h(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) | \boldsymbol{\phi}(\boldsymbol{x}) \text{ is a polynomial feature mapping of order } d\}$$

目标：从$\mathcal{H}$当中选择一个符合预期表现的假设 hypothesis $h(\boldsymbol{x})$.

对于以上问题，其训练误差为：

$$\hat{\epsilon}(h) = \frac{1}{m}\sum_{i=1}^{m}\big(y_i - h(\boldsymbol{x}_i)\big)^2$$

假设训练集和测试集是来自总数据集$\mathcal{D}$的独立同分布的子集，$\mathcal{D}$有些样本不可见，但可能作为未来的预测样本出现，那么泛化误差可以表示为

$$\epsilon(h) = E_{(\boldsymbol{x},y)\sim\mathcal{D}}[\big(y - h(\boldsymbol{x})\big)^2]$$

在训练过程当中，一个学习算法通过最小化训练误差$\hat{\epsilon}(h)$来得到一个 $\hat{h}$:

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \hat{\epsilon}(h)$$

然而我们从上文的分析可以知道，一个较小的训练误差不代表一个较小的泛化误差。我们除了要较小训练误差意外，还需要减小 generalization gap:

$$\text{gap} = \epsilon(h) - \hat{\epsilon}(h)$$

泛化误差可以写成由偏差和方差组成的形式，从中我们可知如何减小误差 gap.

如果我们从总数据集$\mathcal{D}$当中随机选取不同的训练集，那么得到的假设方程常常有所不同，因而他们的泛化误差也有所不同。可以看到，$\hat{h}$和对应的泛化误差$\hat{\epsilon}(h)$是随机取决于训练集S的选取的。因而重写$\hat{h}$为h_S，以表示假设函数仅仅取决于训练集S。

一个好的学习算法需要在所有的训练集，而不仅在一个训练集上拥有较好的表现。所以我们需要最小化的不仅是单独某个泛化误差，而是针对所有训练集的泛化误差的期望 expected generalization error:

$$\epsilon = E_S[\epsilon(h_s)] = E_S[E_{(x,y)\sim\mathcal{D}}[(y - h(\boldsymbol{x}))^2]]$$

拆分以上方程：

$$\epsilon = E_S[\epsilon(h_s)] = E_S\left[E_{(x,y)\sim\mathcal{D}}\left[(y - h(\boldsymbol{x}))^2\right]\right]$$

$$= E_S E_{(x,y)}\left[(y - h_S(\boldsymbol{x}))^2\right]$$

$$= E_S E_{(x,y)}[(y - h_S)^2] \text{ (drop } x \text{ first for simplicity and add it back at the end)}$$

$$= E_S E_{(x,y)}[(y - E_S(h_S) + E_S(h_S) - h_S)^2]$$

$$= E_S E_{(x,y)}\left[(y - E_S(h_S))^2\right] + E_S E_{(x,y)}[(E_S(h_S) - h_S)^2]$$

$$+ 2E_S E_{(x,y)}[((y - E_S(h_S))(E_S(h_S) - h_S)]$$

$$= E_S E_{(x,y)}\left[(y - E_S(h_S))^2\right] + E_S E_{(x,y)}[(E_S(h_S) - h_S)^2]$$

$$+ 2E_{(x,y)}[((y - E_S(h_S))(E_S(E_S(h_S)) - E_S(h_S))] \text{ (red part is 0)}$$

$$= E_S E_{(x,y)}\left[(y - E_S(h_S))^2\right] + E_S E_{(x,y)}[(E_S(h_S) - h_S)^2]$$

$$= E_{(x,y)}\left[(y - E_S(h_S(\boldsymbol{x})))^2\right] + E_S E_{(x,y)}[(E_S(h_S(\boldsymbol{x})) - h_S(\boldsymbol{x}))^2]$$

上式当中，第一项

$$E_{(x,y)}\left[(y - E_S(h_S(\boldsymbol{x})))^2\right]$$

是为偏差的平方 bias²；

第二项

$$E_S E_{(x,y)}[(E_S(h_S(\boldsymbol{x})) - h_S(\boldsymbol{x}))^2]$$

称为方差 variance.

因而有

$$\text{Expected Generalization Error} = \text{Bias}^2 + \text{Variance}$$

偏差是来源于模型本身的错误假设 erroneous assumption. 高的偏差意味着模型做出的分布假设与数据本身的分布有着较大差别，从而会导致特征与目标输出之间的相关关系被忽略，从而造成欠拟合。

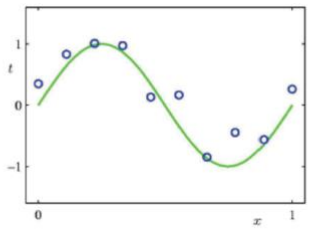
方差代表着模型对训练集中细小的波动的敏感程度 sensitivity to small fluctuations in the training set. 高的方差意味着模型将一些并不是目标输入的随机噪声也考虑进了判别模型当中，从而导致过拟合。

我们要选择一个合适的h，使得偏差和方差都较小。

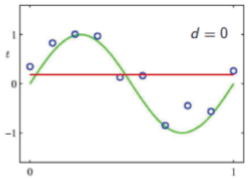
考虑h来自于$\mathcal{H}$

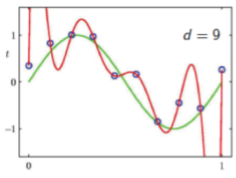
$$\mathcal{H} = \{h(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) | \boldsymbol{\phi}(\boldsymbol{x}) \text{ is a polynomial feature mapping of order } d\}$$

d是超参数。不同的d对偏差和方差的影响如下：



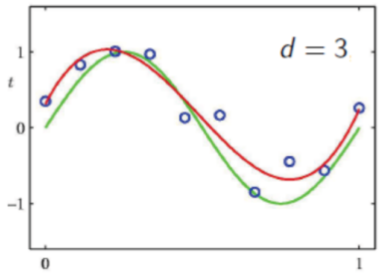
- Suppose the green curve is the true function.
- We randomly sample 10 training points (blue) from the function.
- Consider learning a polynomial function $y = h(x)$ of order d from the data.
- We the above multiple times.





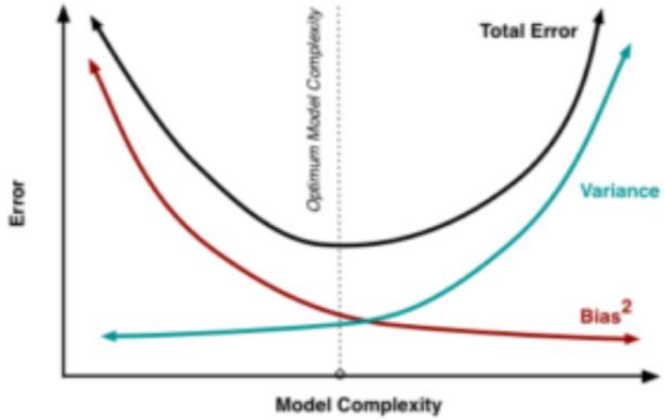
- If we choose $d = 0$, then we have
  - **Low variance**: If there is another training set sampled from the true function (blue) and we run the learning algorithm on it, we will get roughly the same function.
  - **High bias**: While the hypothesis is linear, the true function is not. If we sample a large number of training sets from the true function and learn a function from each of them, the average will still be very different from the true function.
  - In this case, the generalization would be high. And it is due to **underfitting**: hypothesis function too rigid to fit the data points.

- If we choose $d = 9$, then we
  - **High variance**: If there is another training set sampled from the true function and we run the learning algorithm on it, we are likely to get a very different function.
  - **Low bias**: If we sample a large number of training sets from the true function and learn a function from each of them, the average will still approximate the true function well.
  - In this case, the generalization would be high. It is due to **overfitting**: hypothesis too soft, fit the data points too much.



- If we choose $d = 3$, we get low generalization error
  - not too much variance and not too much bias
  - the hypothesis fit the data just right

从以上例子可以看出，偏差常随着模型容量/模型复杂度的增长而减小，而方差正好相反。为了做出较好的决策，需要者这两者之间做出 trade-off，选择一个复杂度适中的模型。



实际上，之前介绍过的交叉验证 Cross Validation 是一个较好的方法。
同时，脊回归 Ridge Regression:

$$J(\boldsymbol{w}, w_0) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x_i})))^2 + \lambda||\boldsymbol{w}||_2^2$$

以及 LASSO:

$$J(\boldsymbol{w}, w_0) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x_i})))^2 + \lambda\big|\big|\boldsymbol{w}\big|\big|_1$$

都是加入了偏差项来得到一个比普通最小二乘解 OLS 更好的表现。

以上的 Bias-Variance 分解是针对最小二乘法来优化的，对于零一损失函数的分类，也可以找到类似的分解方法。

分类问题：VC Dimension

二分类问题描述：给定训练集$S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, 其中$y_i \in \{0,1\}$, 其对应的分类器所组成的假设空间 hypothesis class $\mathcal{H}$：

$$\mathcal{H} = \{h(\boldsymbol{x}, \boldsymbol{w}, b) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + b)|\boldsymbol{w}, b\}$$

目标：从$\mathcal{H}$当中选择一个符合预期表现的假设 hypothesis $h(\boldsymbol{x}, \boldsymbol{w}, b)$.

对于以上问题，其训练误差为：

$$\hat{\epsilon}(h) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\big(y_i \neq h(\boldsymbol{x}_i)\big)$$

假设训练集和测试集是来自总数据集$\mathcal{D}$的独立同分布的子集，$\mathcal{D}$有些样本不可见，但可能作为未来的预测样本出现，那么泛化误差可以表示为

$$\epsilon(h) = E_{(x,y)\sim\mathcal{D}}[\mathbb{I}\big(y \neq h(\boldsymbol{x})\big)]$$

在训练过程当中，一个学习算法通过最小化训练误差$\hat{\epsilon}(h)$来得到一个 $\hat{h}$:

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \hat{\epsilon}(h)$$

我们从 2.2 当中知道，线性分类器面临的凸优化问题常常通过代理损失函数来求解，而非直接通过最小化训练误差。然而以上定义依然有效。

定义最佳假设 optimal hypothesis:

$$h^* = \arg\min_{h\in\mathcal{H}} \epsilon(h)$$

该最佳假设是最小化泛化误差的假设，然而我们无法直接得到这个假设。

引入概念：我们称二分类器所组成的假设空间 hypothesis class $\mathcal{H}$打碎 shatters 了一个数据集，假若其满足以下条件：即不论我们如何给数据集打上 0 或 1 的标签，总有一个$h \in \mathcal{H}$可以零失误的将数据集分成两类。

一个二分类假设空间$\mathcal{H}$的 Vapnik-Chervonenkis/VC dimension $VC(\mathcal{H})$为其能 shatter 的最大数据集的 size；若一个$\mathcal{H}$可 shatter 任意大小的集，则$VC(\mathcal{H}) = \infty$

Example: $\mathcal{H}_e = \{\text{Linear Classifiers with two input } x_1 \text{ and } x_2\}$

$\mathcal{H}_e$可最少 shatter 一个 3 个输入点的数据集，但不能 shatter 任意一个 4 输入点的数据集，因此$VC(\mathcal{H}_e) = 3$.

Vapnik-Chervonenkis Theorem:

给定训练集$S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, 其中$y_i \in \{0,1\}$, 其对应的分类器所组成的假设空间 hypothesis class $\mathcal{H}$, 令$d = VC(\mathcal{H})$, 则有

1. 对于任何$h \in \mathcal{H}$, 以下式子发生概率至少为$1-\delta$:

$$|\epsilon(h) - \hat{\epsilon}(h)| \leq O(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}})$$

2. 以下式子发生概率至少为$1-\delta$:

$$\epsilon(\hat{h}) \leq \hat{\epsilon}(h^*) + O(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}})$$

其中，$\epsilon(h)$为训练误差 training error；$\hat{\epsilon}(h)$为泛化误差 generalization error；$\epsilon(\hat{h})$最小训练误差；$\hat{\epsilon}(h^*)$为最小泛化误差。

由 VC 理论，我们可以知道：

1. 根据 VC 理论的第一条，任意假设方程h的训练误差$\epsilon(h)$和泛化误差$\hat{\epsilon}(h)$是紧密相连的，因此降低泛化误差$\hat{\epsilon}(h)$的同时也可以降低训练误差$\epsilon(h)$；同时，样本大小m越大，训练误差$\epsilon(h)$和泛化误差$\hat{\epsilon}(h)$的联系就越紧密，因而越多的样本可以给模型带来更好的表现；

   以下的推论 Corollary 说明了到底需要多少数据方可使一个算法表现优异：令

$$\gamma = O(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}})$$

   若要让任何$h \in \mathcal{H}$, 以下式子发生概率至少为$1-\delta$:

$$|\epsilon(h) - \hat{\epsilon}(h)| \leq \gamma$$

   则m应满足：

$$m = O_{\gamma,\delta}(d)$$

2. VC 理论的第二部分提供了一个减小最小训练误差分类器的泛化误差的思路：增大模型复杂度/模型容量会使第二部分右半式第一项偏差$\hat{\epsilon}(h^*)$减小，但使第二项方差$O(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}})$增大；增大数据集大小则会使第二项减小，从而避免过拟合；Validation 和正则化是两个提供合适 trade-off 的方法。

   关于多少数据才可使最小训练误差分类器$\hat{h}$有接近最优的泛化误差：有推论：若要让任何$h \in \mathcal{H}$, 以下式子发生概率至少为$1-\delta$:

$$\epsilon(\hat{h}) \leq \hat{\epsilon}(h^*) + 2\gamma$$

   则m应满足$m = O_{\gamma,\delta}(d)$.