第7章　统计学习的本质

刘家锋

哈尔滨工业大学

第7章　统计学习的本质

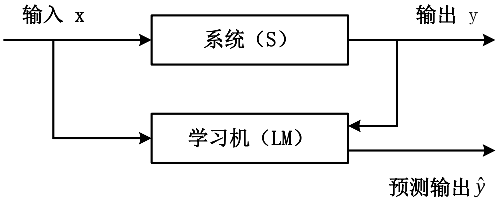
7.1 统计学习的本质

分类器的泛化能力

- **分类器学习的目的**
  - 在训练样本集上设计和学习分类器的目的，是对未来的测试样本进行分类
  - 如果分类器只能正确分类训练样本，而对测试样本的分类错误率高，则称分类器的泛化性能差
  - 提高分类器的泛化性能是学习的根本目的
- **机器学习的本质是什么？**
  - 是什么原因导致分类器的泛化能力差？
  - 如何来提高分类器学习的泛化能力？

机器学习过程

- **学习的过程**
  - 以系统S为研究对象，输入矢量$\mathbf{x}$，输出y
  - 学习机LM通过一系列的观察样本$\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$，来模仿系统S
  - 输入$\mathbf{x}$时，LM的输出$\hat{y}$能够尽量准确地预测S的输出y



机器学习问题

- **根据输出y的不同，学习问题分为两类**
  - 分类问题：离散值输出，$y \in \{1, \cdots, c\}$
  - 回归问题：连续值输出，$y \in R$
- **学习机**
  - 学习机LM的输出$\hat{y}$与输入$\mathbf{x}$可以看作是函数关系：$\hat{y} = f(\mathbf{x})$
  - 例如线性的函数关系
    - 线性回归：$\hat{y} = \mathbf{w}^t\mathbf{x} + b$
    - 线性分类：$\hat{y} = \begin{cases} +1, & \mathbf{w}^t\mathbf{x} + b \geq 0 \\ -1, & \mathbf{w}^t\mathbf{x} + b < 0 \end{cases}$

学习的风险

- **定义风险**
  - 输入**x**时，系统输出y而学习机输出$\hat{y}$的损失定义为风险

$$L(y, \hat{y}) = L(y, f(\mathbf{x}))$$

- **常用的风险**
  - 平方误差风险

$$L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$$

  - Hinge loss

$$L(y, f(\mathbf{x})) = \max\left(0, 1 - yf(\mathbf{x})\right), \qquad y \in \{-1, +1\}$$

  - 似然函数

$$L(p(\mathbf{x})) = -\ln p(\mathbf{x})$$

期望风险

- **期望风险**
  - 系统S的输出y与输入$\mathbf{x}$之间存在的依赖关系，可以用联合概率分布$F(\mathbf{x}, y)$ 来描述
  - 学习的期望风险为定义在f上的泛函

$$R(f) = \int L\left(y, f(\mathbf{x})\right) dF(\mathbf{x}, y)$$

其中的积分式为Stieltjes积分

  - 机器学习的目标就是要优化期望风险

$$\min_{f} R(f) = \int L\left(y, f(\mathbf{x})\right) dF(\mathbf{x}, y)$$

期望风险

- **参数化的期望风险**
  - 在所有的函数中优化期望风险是不可行的
  - 机器学习一般将优化限定在一组特定的函数中，例如线性函数，高斯函数，GMM，给定结构的神经网络等等
  - 特定的函数可以表示为参数化的形式：$f(\mathbf{x}, \mathbf{w})$
  - 机器学习的目标变成了对参数$\mathbf{w}$的优化

$$\min_{\mathbf{w}} R(\mathbf{w}) = \int L\left(y, f(\mathbf{x}, \mathbf{w})\right) dF(\mathbf{x}, y)$$

经验风险

- **分布的抽样**
  - 联合分布$F(\mathbf{x}, y)$是未知的，期望风险一般是无法计算的
  - 联合分布是可以抽样的，得到样本集

$$\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\} \sim F(\mathbf{x}, y)$$

- **经验风险**
  - 工程上使用经验风险来近似期望风险

$$R(\mathbf{w}) \approx R_{emp}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i, \mathbf{w}))$$

经验风险

- **经验风险的优化**
  - 机器学习的过程转化为对经验风险的优化

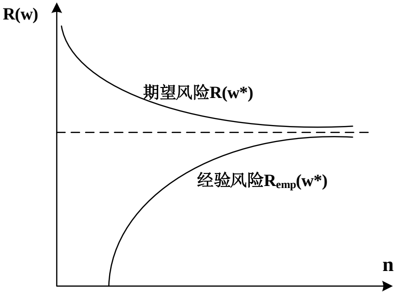
$$\min_{\mathbf{w}} R_{emp}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i, \mathbf{w}))$$

- **学习过程的一致性**
  - 经验风险的优化与期望风险是一致的吗？
  - 在统计学上，关于一致性有如下结论

$$\lim_{n\to\infty} P\left\{\sup_{\mathbf{w}} |R_{emp}(\mathbf{w}) - R(\mathbf{w})| > \epsilon\right\} = 0, \quad \forall \epsilon > 0$$

经验风险与期望风险

- 经验风险与期望风险的关系
  - 经验风险总是小于期望风险的：$R_{emp}(\mathbf{w}) \leq R(\mathbf{w})$
  - 当抽样数$n \to \infty$时，两者趋于一致



经验风险与期望风险

- **机器学习可靠吗？**
  - 所有的机器学习方法，本质上都是用经验风险替代期望风险优化
  - 统计学告诉我们，当训练样本数$n \to \infty$时，替代优化经验风险是可靠的
  - 当样本本数n有限时，优化经验风险还是可靠的吗？
- **机器学习的泛化**
  - 最小化经验风险可以提高训练数据预测的准确率
  - 但当训练样本数不足时，对未知测试样本预测的准确率可能会很低
  - 学习模型的泛化能力可能出现问题

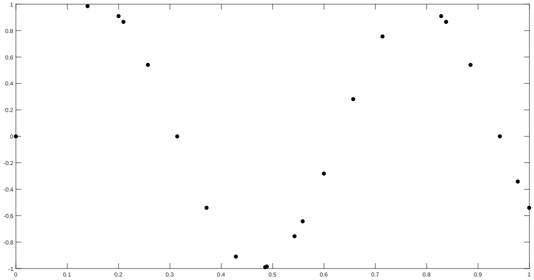
7.2 VC维与泛化界

模型的复杂程度

- **模型的泛化能力**
  - 研究发现，泛化能力不仅与样本数n有关，也与学习机的函数集选择有关
  - “简单”的函数集泛化能力强，“复杂”的泛化能力差
- **过学习(overfitting)**
  - 当函数集过于“复杂”时，很容易产生“过学习”现象
  - 对于训练样本风险很小，而对非训练样本风险却很大

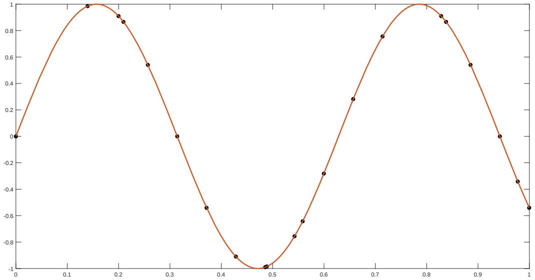
Overfitting

- **过学习现象**
  - 对函数$f(x)$抽样得到一组数据



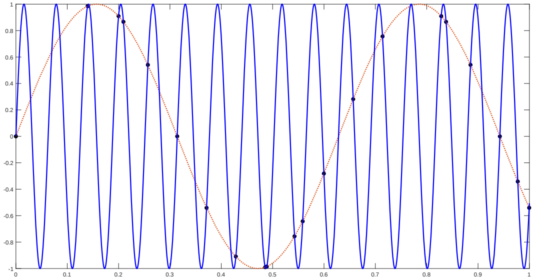
Overfitting

- **过学习现象**
  - 对函数$f(x)$抽样得到一组数据
  - 使用正弦函数集合$\{\sin(\omega x)\}$来拟合数据，ω_*是学习的参数



Overfitting

- **过学习现象**
  - 对函数$f(x)$抽样得到一组数据
  - 使用正弦函数集合$\{\sin(\omega x)\}$来拟合数据，ω_*是学习的参数
  - 优化的解不是唯一的，$\sin(\omega_* x)$可以拟合数据，$\sin(10\omega_* x)$也可以很好地拟合数据

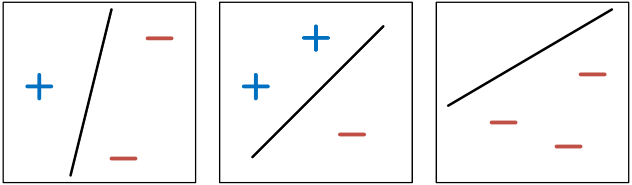


模型的复杂程度

- **定性度量**
  - 定性来说，如果函数集合$S_1 \subset S_2$，则S_2比S_1更复杂
  - 函数集合S_1能够拟合或分类的数据，S_2同样能够拟合或分类，反之则不然
  - 例如，二次函数集合比线性函数集合更复杂，高斯数量多的GMM更复杂，隐含层神经元数量多的神经网络更复杂
- **定量度量**
  - 定量地度量一个函数集合的复杂程度是很困难的
  - Vapnik提出的VC维是度量函数集复杂程度和机器学习模型泛化能力的一个指标

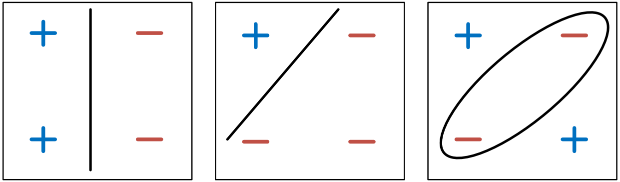
Vapnik-Chervonenkis Dimension

- **打散**
  - 如果存在一个有h个样本的样本集，能够被一个函数集中的函数按照所有可能的2^h种形式分为两类，则称函数集能够将样本数为h的样本集打散
  - 2维空间中
    - 线性函数集合能够打散$h = 3$的样本集，不能打散$h = 4$的样本集
    - 二次函数集合能够打散$h = 4$的样本集合



Vapnik-Chervonenkis Dimension

- **打散**
  - 如果存在一个有h个样本的样本集，能够被一个函数集中的函数按照所有可能的2^h种形式分为两类，则称函数集能够将样本数为h的样本集打散
  - 2维空间中
    - 线性函数集合能够打散$h=3$的样本集，不能打散$h=4$的样本集
    - 二次函数集合能够打散$h=4$的样本集合



Vapnik-Chervonenkis Dimension

- **VC维**
  - 如果函数集能够打散h个样本的集合，而不能打散$h+1$个样本的集合，则称函数集的VC维为h
  - 可以证明，d维空间中线性函数集合的VC维：$h=d+1$
  - 正弦函数集合$\{\sin(\omega x)\}$的VC维：$h=\infty$

VC维与泛化能力

- **期望风险的界**
  - 期望风险无法准确计算，利用VC维和经验风险可以确定期望风险的上界和下界
  - Vapnik证明了，下列不等式成立的概率大于$1-\eta$

$$R_{emp}(\mathbf{w}) \leq R(\mathbf{w}) \leq R_{emp}(\mathbf{w}) + \sqrt{\frac{h(\ln(2n/h)+1)-\ln(\eta/4)}{n}}$$

其中，n为样本数，h为函数集合的VC维，$1-\eta$是上界的置信度

  - 忽略置信度，不等式可以简写为

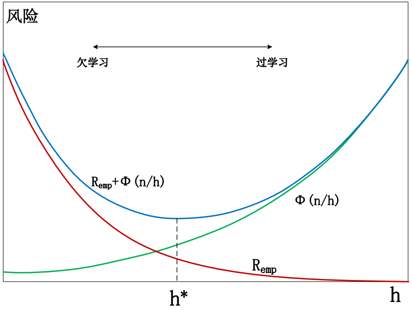
$$R_{emp}(\mathbf{w}) \leq R(\mathbf{w}) \leq R_{emp}(\mathbf{w}) + \Phi(n/h)$$

  - n/h越大上界越紧，越小上界越松

7.3 提高泛化能力的方法

结构风险最小化

- **优化期望风险的上界**
  - 优化期望风险的下界$R_{emp}(\mathbf{w})$，过于乐观，容易过学习
  - 选用简单模型优化$\Phi(n/h)$，经验风险$R_{emp}(\mathbf{w})$大，欠学习
  - 优化期望风险的上界$R_{emp}(\mathbf{w})+\Phi(n/h)$，兼顾经验风险和模型的复杂度，可提高学习的泛化能力



结构风险最小化

- **Structural Risk Minimization**
  - 将函数集合$S=\{f(\mathbf{x},\mathbf{w})\}$分解为子集的序列

$$S_1 \subset S_2 \subset \cdots S_k \subset \cdots \subset S$$

    对应的VC维：$h_1 \le h_2 \le \cdots \le h_k \le \cdots$
  - 在序列中寻找使得期望风险上界最小的子集S_*
  - 在子集S_*中优化经验风险，得到的函数为最优函数

$$\mathbf{w}_* = \arg\min_{\mathbf{w}} R_{emp}(\mathbf{w}), \quad s.t. \quad f(\mathbf{x},\mathbf{w}) \in S_*$$

线性分类器上的结构风险最小化

Theorem

令包含d维空间中n个训练样本超球体的半径为R，γ-间隔分类超平面集合的*VC*维h满足：

$$h \leq \min\left(\frac{R^2}{\gamma^2}, d\right) + 1$$

- Vapnik证明当限制判别界面的分类间隔时，d维空间中线性函数的VC维可以小于$d+1$
- 当超平面能够正确分类所有训练样本时，经验风险为0，最大化间隔$\gamma = 1/\|\mathbf{w}\|$可以降低线性分类器的VC维，提高泛化能力

线性分类器上的结构风险最小化

- **支持向量机**
  - SVM的优化依据的就是结构风险最小化原则

$$\min_{\mathbf{w},w_0} J(\mathbf{w},w_0)=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\max[0,1-y_i(\mathbf{w}^t\mathbf{x}_i+w_0)]$$

    - 第一项优化的是分类间隔γ，即模型的VC维
    - 第二项优化的是经验风险，表示为Hinge loss形式
- **最小二乘支持向量机**
  - 将经验风险替换为平方误差损失，得到最小二乘支持向量机(LS-SVM)的优化目标

$$\min_{\mathbf{w},w_0} J(\mathbf{w},w_0)=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}(y_i-(\mathbf{w}^t\mathbf{x}_i+w_0))^2$$

其它方法

- **权值衰减**
  - 计算和优化神经网络的VC维是很困难的
  - 实践表明，控制网络权值的大小，可以提高泛化能力
  - 在优化平方误差$J(\mathbf{w})$的基础上，同时优化权值矢量的长度$\|\mathbf{w}\|$，可以达到结构风险最小化的目的

$$J_{wd}(\mathbf{w}) = J(\mathbf{w}) + \frac{\epsilon}{2\eta}\|\mathbf{w}\|^2$$

  其中，η为学习率，ϵ为权值衰减系数

  - 可以证明，相应的梯度迭代公式为

$$\mathbf{w} \leftarrow (1-\epsilon)\mathbf{w} - \eta\nabla J(\mathbf{w})$$

验证技术

- **Validation**
  - 将数据集D划分为不相交的两部分：$D = D_t \cup D_v$
  - 使用训练集D_t学习分类器，由验证集D_v决定何时停止
  - 在神经网络的学习中，这种方法也称为“提前停止”