机器学习与数据科学

Machine Learning and Data Science

主讲: 李春光

www.pris.net.cn/teacher/lichunguang

模式识别与智能系统实验室

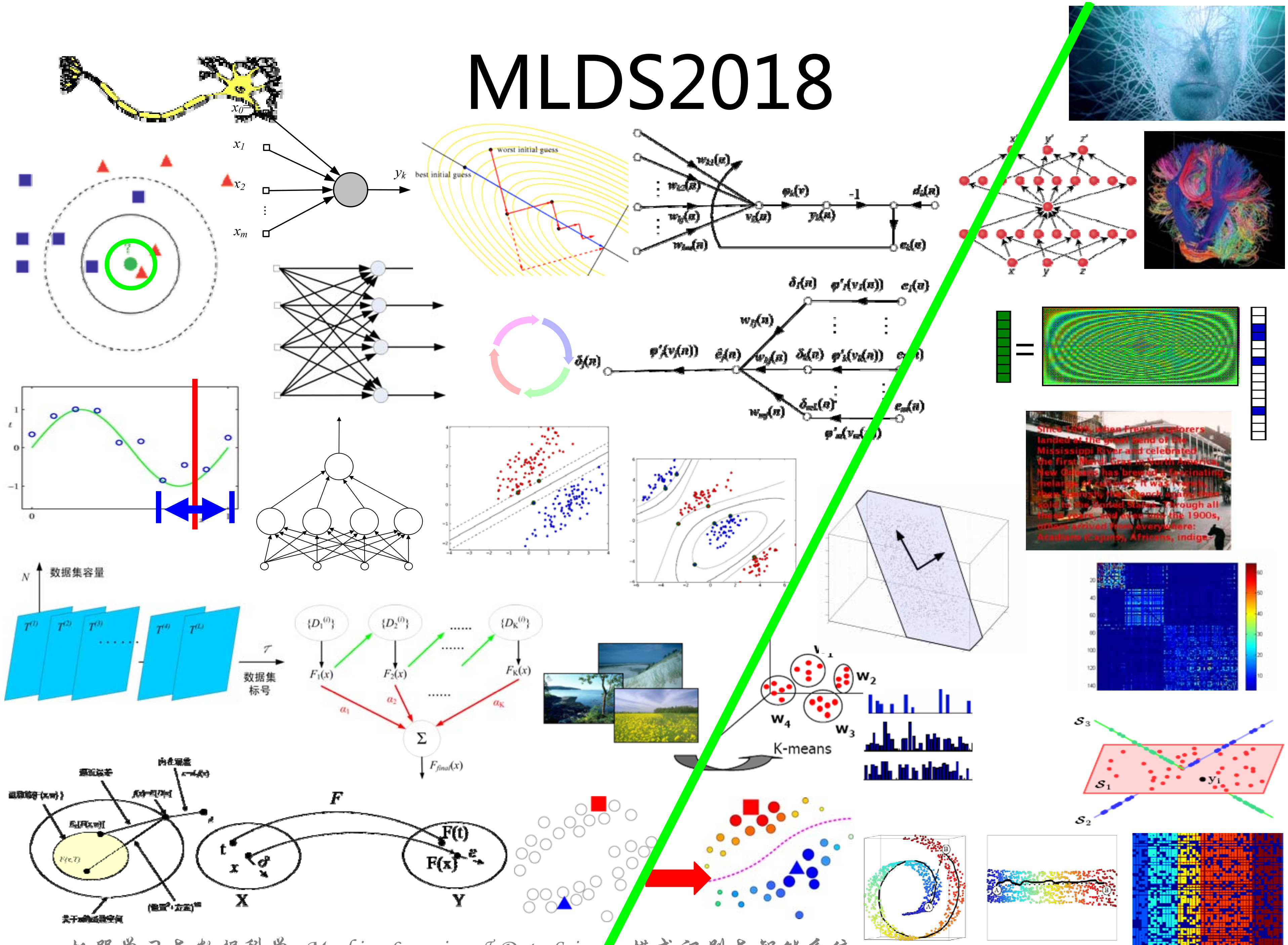
信息与通信工程学院 网络搜索教研中心

北京邮电大学

专题 五：学习过程的统计性质与集成学习

- **内容提要**
  - 学习过程的统计性质
    - 误差分解
    - 偏倚/方差困境
  - 集成学习
    - 平均法
    - 推举法
    - 混合专家

MLDS2018

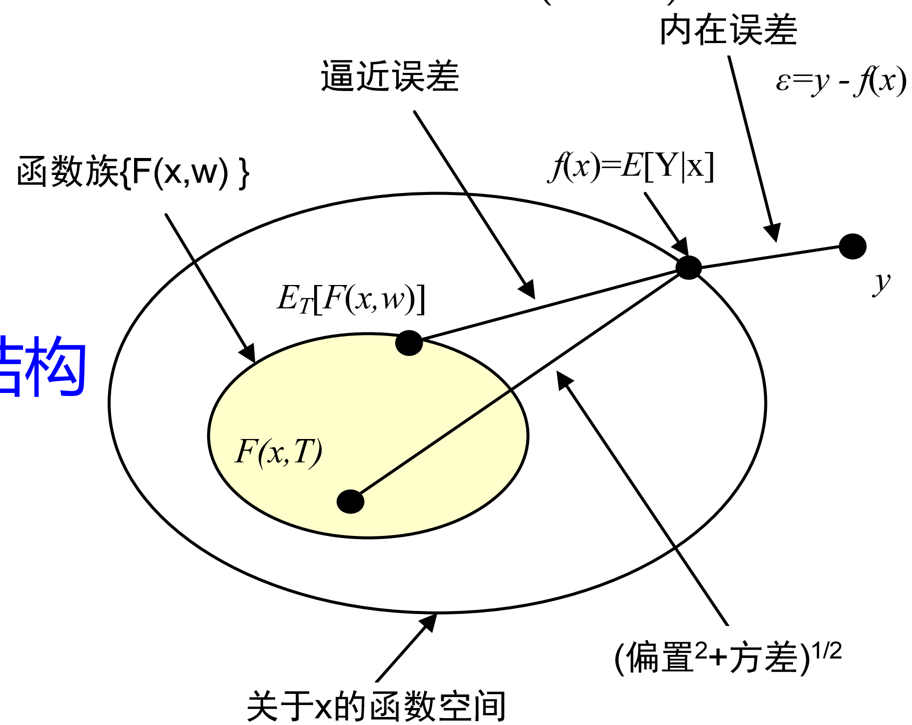


机器学习与数据科学 - *Machine Learning & Data Science* 模式识别与智能系统实验室

- **内容提要**
  - 学习过程的统计性质
    - 误差分解
    - 偏倚/方差困境
  - 集成学习
    - 平均法
    - 推举法
    - 混合专家

引言

- 在本专题中，我们以回归模型为例，分析误差的来源
  - 我们所关心的不是权值 $\boldsymbol{w}$ 的更新, 而是目标函数 $f(\mathbf{x})$ 和由回归模型所实现的函数 $F(\mathbf{x}, \mathbf{w})$ 之间的误差
  - 主要目标:
    - 看懂右图所示“鸡蛋”结构



回归模型的统计表达

- 回归模型
  - 考虑一个物理过程的观测随机向量X和随机标量Y，假设经过N次测量得到训练样本集：

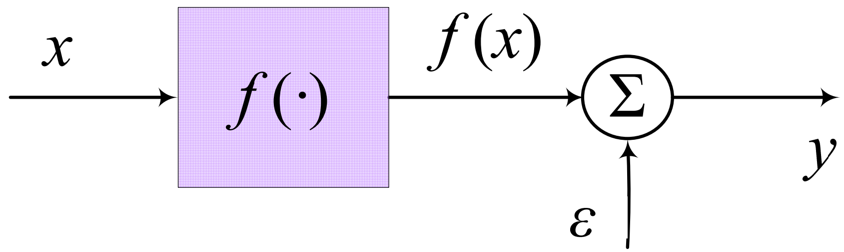
$$T=\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{i=1}^{N}$$

  - 对于Y和X之间的函数关系提出模型：

$$Y=f\left(X\right)+\varepsilon$$

    - 其中f(.)表示一个确定性函数，ε表示随机误差

- 回归模型两个基本假设：



  - 误差零均值： $\mathbf{E}\left[\varepsilon \mid X=\mathbf{x}\right]=0$
    - 回归函数f(X)是在给定输入X=x的情况下模型输出Y的条件均值，即

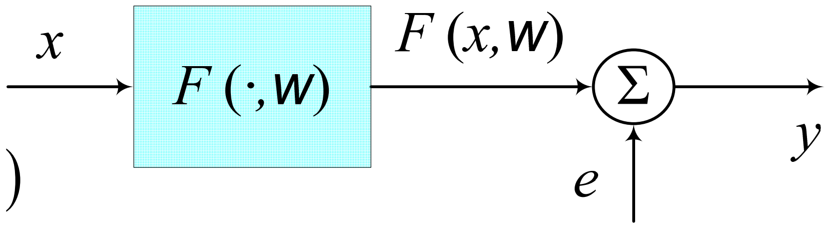
$$f\left(\mathbf{x}\right)=\mathbf{E}\left[Y \mid X=\mathbf{x}\right]$$

  - 误差ε和回归函数 f(X) 不相关

$$\mathbf{E}\left[\varepsilon f\left(X\right)\right]=0$$

以神经网络为例

- 基于神经网络的回归模型



$$Y = F\left(\mathbf{X}, \mathbf{w}\right)$$

其中，$F\left(\cdot, \mathbf{w}\right)$表示由神经网络实现的输入–输出函数

  - **神经网络提供对数学回归模型的近似**

- **神经网络把由训练样本集T所表示的经验知识编码到对应的权值向量w中，两种写法F(x,T)与F(x,w)等价**

  - **定义代价函数为：**

$$\tau\left(\mathbf{w}\right) = \int\int\left[y - F\left(\mathbf{x}, \mathbf{w}\right)\right]^2 p\left(\mathbf{x}, y\right) d\mathbf{x} dy = \mathbf{E}\left\{\left[Y - F\left(X, \mathbf{w}\right)\right]^2\right\}$$

期望算子E[·]作用在随机变量X和Y的总体上

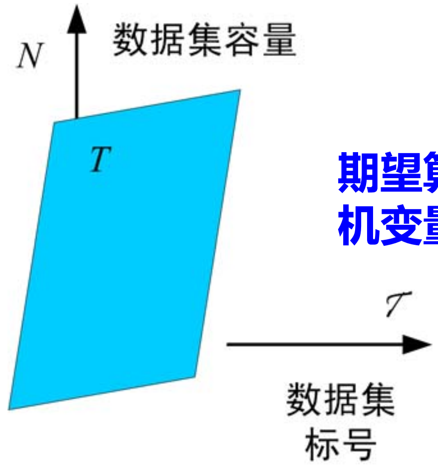
    - **在训练数据集T上的误差：**

$$\overline{\tau}\left(\mathbf{w}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - F\left(\mathbf{x}_i, \mathbf{w}\right)\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - F\left(\mathbf{x}_i, \mathbf{T}\right)\right)^2$$

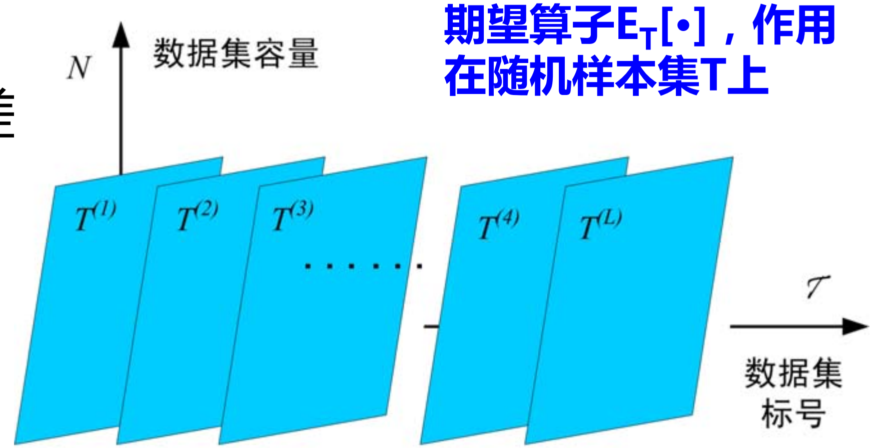
误差分解的两个层次图示

$$\tau(\mathbf{w})=\iint\left[y-F(\mathbf{x},\mathbf{w})\right]^2 p(\mathbf{x},y)\,d\mathbf{x}dy=\mathbf{E}\left\{\left[Y-F(X,\mathbf{w})\right]^2\right\}=\mathbf{E}\left\{\left[Y-F(X,\mathbf{T})\right]^2\right\}$$

- 在一个数据集(总体)T上进行的“纵向”误差分解
  - **第一层分解**
  - **误差 = 内在误差 + 估计误差**
- 在多个数据集{$T^{(i)}$}上进行的“横向”误差分解
  - **对于估计误差进行第二层分解**
  - **估计误差$|_x$ = (偏倚^2 + 方差)$^{1/2}$**



期望算子E[•]，作用在随机变量X和Y的总体上



期望算子E_T[•]，作用在随机样本集T上

误差分解的第一层次

- 误差 = 内在误差 + 估计误差

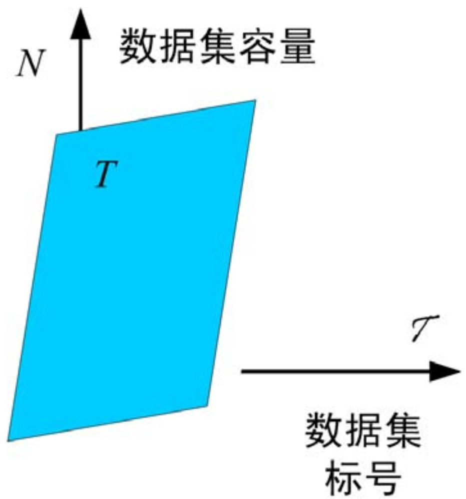
$$\tau(\mathbf{w}) = \mathbf{E}\left\{\left[Y - F(X,\mathbf{w})\right]^2\right\} = \mathbf{E}\left\{\left[\left(Y - f(X)\right) + \left(f(X) - F(X,\mathbf{w})\right)\right]^2\right\}$$

$$= \mathbf{E}\left[\varepsilon^2\right] + \mathbf{E}\left[\left(f(X) - F(X,\mathbf{w})\right)^2\right] + \underline{2\mathbf{E}\left[\varepsilon \cdot \left(f(X) - F(X,\mathbf{w})\right)\right]}$$

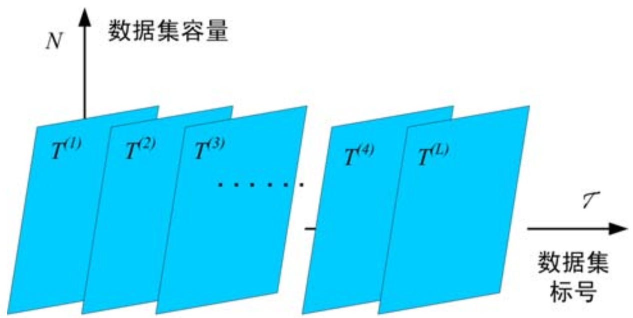
$$= \mathbf{E}\left[\varepsilon^2\right] + \mathbf{E}\left[\left(f(X) - F(X,\mathbf{w})\right)^2\right]$$

其中，交叉项为0，原因: (1) 正交性原理；(2) ε来自理论回归模型，F(x,T)来自实际神经网络模型

- **第1项：$\mathbf{E}\left[\varepsilon^2\right]$**
  - **回归模型误差ε的方差**
  - **代表内在误差，独立于权值向量**
- **第2项：$\mathbf{E}\left[\left(f(X) - F(X,\mathbf{w})\right)^2\right]$**
  - **回归函数和逼近函数之间的期望平方距离**
  - **衡量F(x,w)用于预测 f(x) 的有效性**



误差分解的第二层次



- 对于 X=x 处的估计误差，进行第二层次的分解
  - **X=x 处的估计误差**

$$\left(f(\mathbf{x})-F(\mathbf{x},T)\right)^2 \quad =\left(f(\mathbf{x})-\bar{F}(\mathbf{x})+\bar{F}(\mathbf{x})-F(\mathbf{x},T)\right)^2$$

  - **相对于样本集 T 取期望E_T：**

其中， $\bar{F}(\mathbf{x})=E_T\left[F(\mathbf{x},T)\right]$

$$\begin{aligned}
E_T\left[\left(f(\mathbf{x})-F(\mathbf{x},T)\right)^2\right] &= E_T\left\{\left[f(\mathbf{x})-\bar{F}(\mathbf{x})\right]^2\right\}+E_T\left\{\left[\bar{F}(\mathbf{x})-F(\mathbf{x},T)\right]^2\right\} \\
&=\left[f(\mathbf{x})-\bar{F}(\mathbf{x})\right]^2+E_T\left\{\left[F(\mathbf{x},T)-\bar{F}(\mathbf{x})\right]^2\right\} \\
&= B^2(\mathbf{x})+V(\mathbf{x})
\end{aligned}$$

- **第1项B(x)：F(x,T)的平均值对于回归函数 f(x) 的偏倚**
  - 看作逼近误差，B(x)>0说明由 F(x,T) 所定义的神经网络不能精确逼近回归函数 f(x)
- **第2项V(x) ：在一组训练样本集{$T^{(i)}$}上测量的逼近函数F(x,w)的方差**
  - 体现 F(x,T) 随着样本集T的变化而产生的变异程度

误差的累积过程

- **内在误差定义为 (对数据总体)**

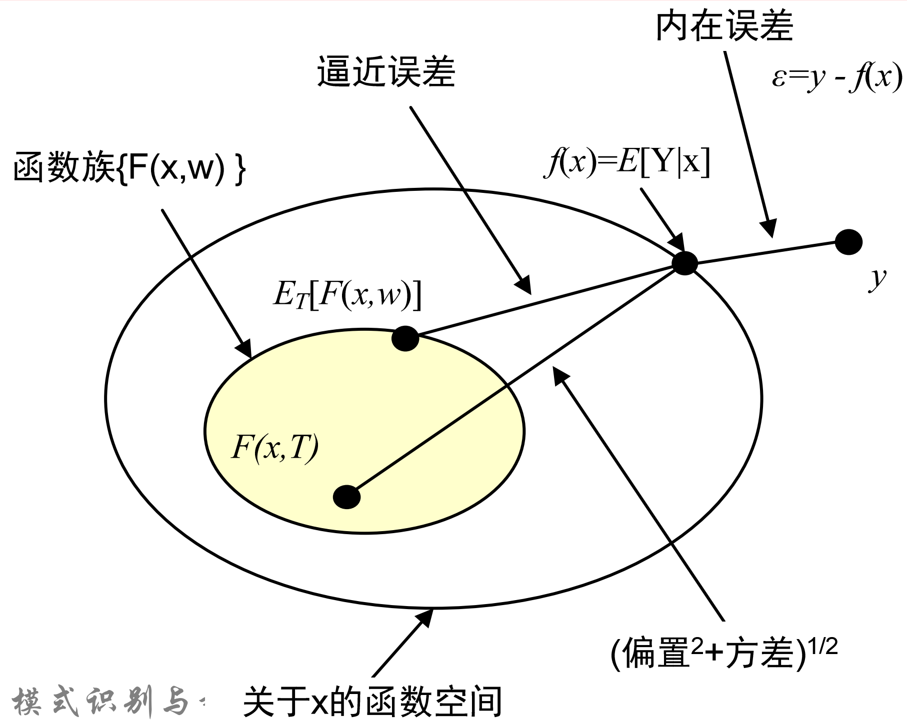
$$\tau(\mathbf{w}) = \mathbf{E}\left\{\left[Y - F(X,\mathbf{w})\right]^2\right\} = \mathbf{E}\left[\varepsilon^2\right] + \mathbf{E}\left[\left(f(X) - F(X,\mathbf{w})\right)^2\right]$$

- **学习机器的逼近误差**

$$E_T\left[\left(f(\mathbf{x}) - F(\mathbf{x},T)\right)^2\right] = \left[f(\mathbf{x}) - \bar{F}(\mathbf{x})\right]^2 + E_T\left\{\left[F(\mathbf{x},T) - \bar{F}(\mathbf{x})\right]^2\right\}$$
$$= B^2(\mathbf{x}) + V(\mathbf{x})$$

  - **要减小整体误差，需要减少逼近函数F(x,w)的偏倚和方差**

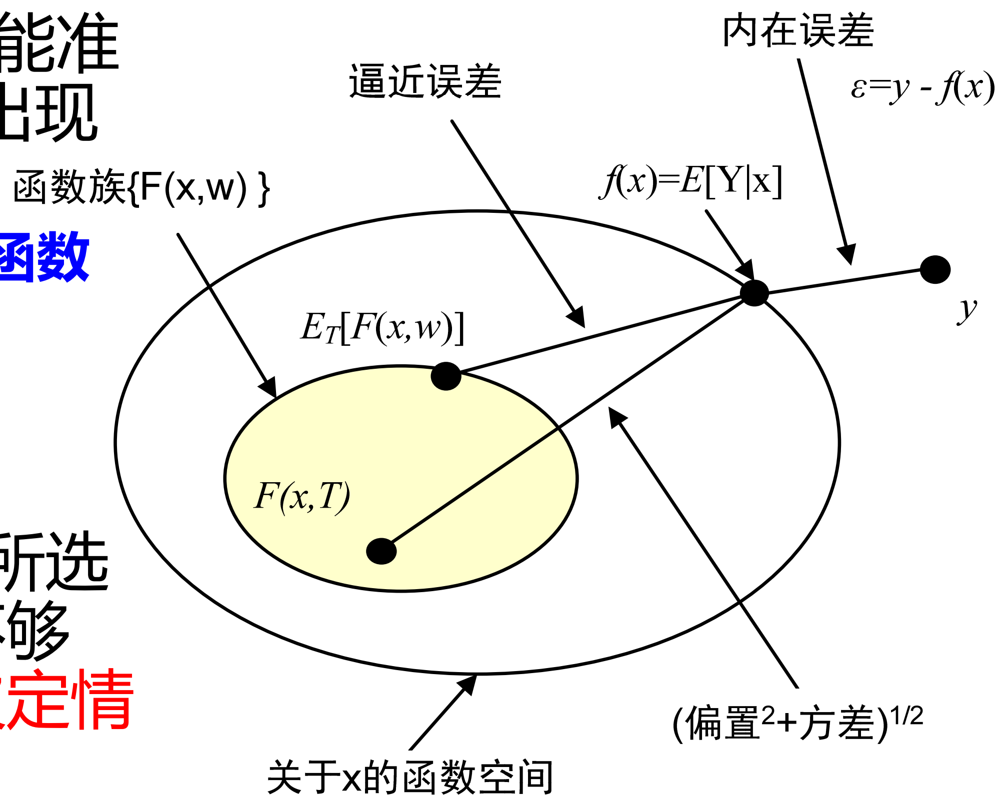
理解为: (1) 在x 处的截面图; (2) 不同的函数空间



偏倚/方差的存在原因

- 偏倚的存在原因
  - 所选择的模型F(x,w)不能准确逼近回归函数f(x)，出现学习问题的过定情况
    - **模型过于简单，如线性函数**
      - **出现欠学习**

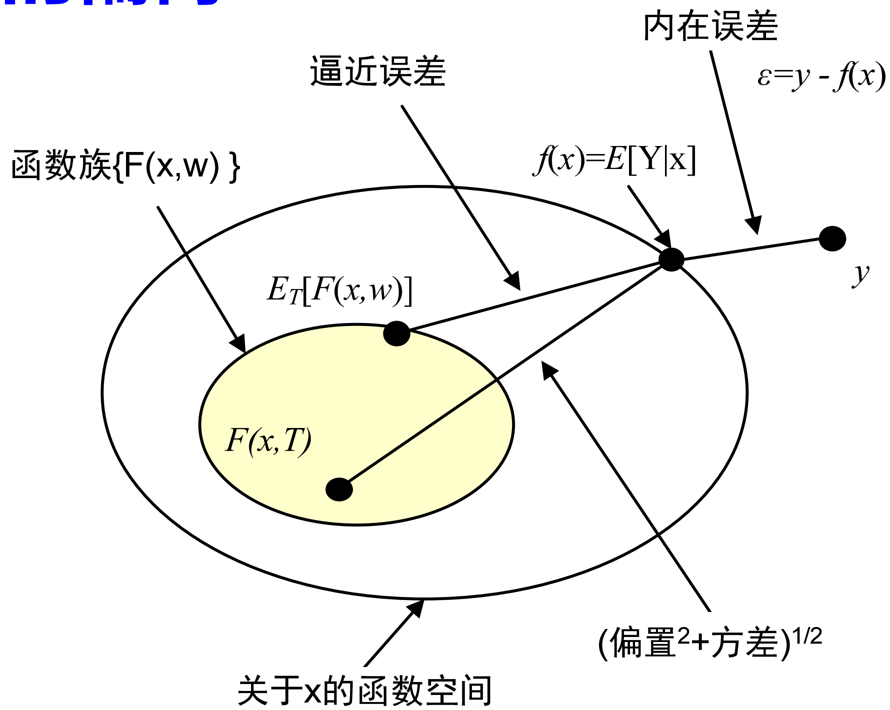
- 方差的存在原因
  - 给定的训练样本集T与所选择函数集的容量相比不够大，出现学习问题的欠定情况
    - **与模型复杂度相比，训练样本不足**
      - **出现过学习**



偏倚/方差困境

- 偏倚/方差困境
  - **在固定大小的训练集上训练单个学习模型，获得较小的偏倚往往会使得方差增大，获得较小的方差往往会带来较大的偏倚**

- 如何获得小的偏倚?
- 如何获得小的方差?



从偏倚/方差分解角度看算法

- 举例

  1. 线性模型, 比如最小二乘线性回归

$$F\left(\mathbf{x},\mathbf{w}\right)=\mathbf{w}^{T}\cdot\mathbf{x}$$

  - **由于全局线性约束，因此方差小，偏倚大**

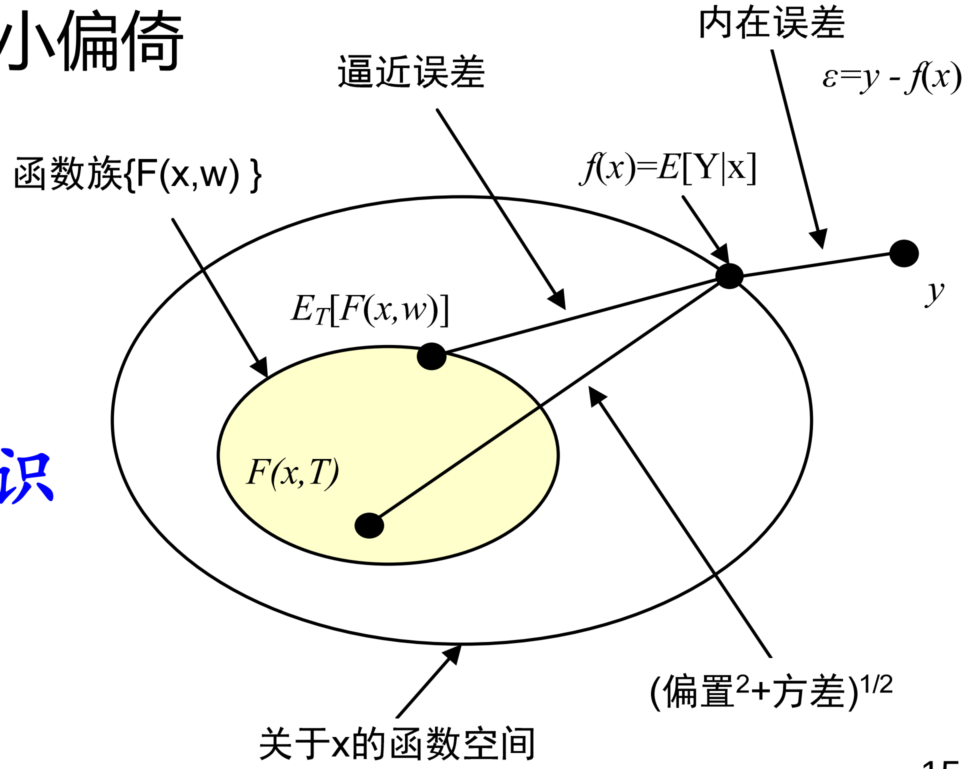
  2. K近邻模型，比如k近邻回归

$$F\left(\mathbf{x},k\right)=\frac{1}{k}\sum_{j\in N\left(\mathbf{x}\right)}y_{j}$$

  - **是对** $f\left(\mathbf{x}\right)=E\left[Y\mid X=\mathbf{x}\right]$ **的两次近似**
  - **仅包含局部上的少量约束，因此偏倚小，方差大**

偏倚/方差分解的指导意义

- 偏倚方差困境的应对策略
  - 对于固定容量的数据集，要减小学习机器的期望误差，需要在偏倚和方差之间寻找折中
    - 通过引入偏倚，来消减方差
    - 通过增加方差，来减小偏倚



- **破解之道：**
  1. 增加训练数据
  2. 加入问题的先验知识

思考问题: 为什么呢？

Q / A

- Any Question? …

专题 五：学习过程的统计性质与集成学习

- **内容提要**
  - 学习过程的统计性质
    - 误差分解
    - 偏倚/方差困境
  - 集成学习
    - 平均法
    - 推举法
    - 混合专家

引言

- 作为学习过程的统计性质的直接应用，我们介绍用于减小误差的通用策略
  - 集成学习
    - **把复杂的任务分解成一组简单的任务，再把这些任务的解重新组合起来**
      - **专家: 用于解决特定任务的单个模型**
      - **委员会：多个专家的组合体**
    - 集成学习也叫模型组合

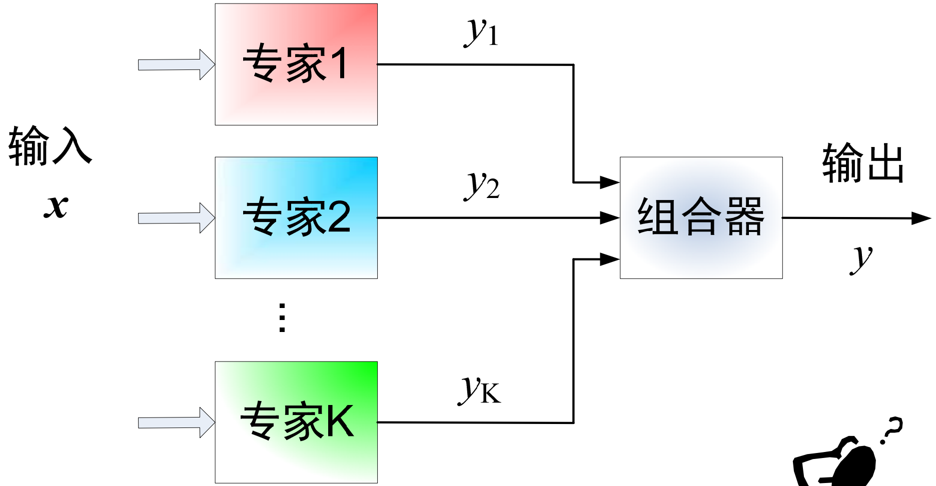
专题 五：学习过程的统计性质与集成学习

- **内容提要**
  - 学习过程的统计性质
    - 误差分解
    - 偏倚/方差困境
  - 集成学习
    - 平均法
    - 推举法
    - 混合专家

总体平均

- 基本思想
  - **分别并行训练多个专家**
    - **比如：使用数据集的切片**(Bootstrap)**, 或使用不同初始值，收敛到误差曲面的不同局部极小**
  - **通过对多个输出的某种组合而提高整个系统性能**
- 集成方法：
  - 回归任务
    - **相加求平均**

$$y=\frac{1}{K}\sum_{i=1}^{K}y_i$$



  - 分类任务
    - **投票法 (绝对多数或相对多数)**

从偏倚-方差分解看总体平均法

- 利用不同初始值或数据的不同子集的MLP训练出不同的专家
  - **假设K个初始权值,对应获得K个专家** $y_k(x), k=1,\ldots,K$ **则**
  - **假设** $y_k(x)=h(x)+\varepsilon_k,$

$$E_{AV}=\frac{1}{K}\sum_{k=1}^{K}E_x\left[\varepsilon_k^{\ 2}\right]$$ ，其中 $$E_x\left[\left\{y_k(x)-h(x)\right\}^2\right]=E_x\left[\varepsilon_k^{\ 2}\right]$$

$$E_{COM}=E_x\left[\left\{y_{COM}(x)-h(x)\right\}^2\right]=E_x\left[\left\{\frac{1}{K}\sum_{k=1}^{K}y_k(x)-h(x)\right\}^2\right]$$

$$=E_x\left[\left\{\frac{1}{K}\sum_{k=1}^{K}\varepsilon_k\right\}^2\right]$$ 其中 $$y_{COM}(x)=\frac{1}{K}\sum_{k=1}^{K}y_k(x)$$

  - **假设：误差** ε_k **均值为0，且相互不相关，则：**

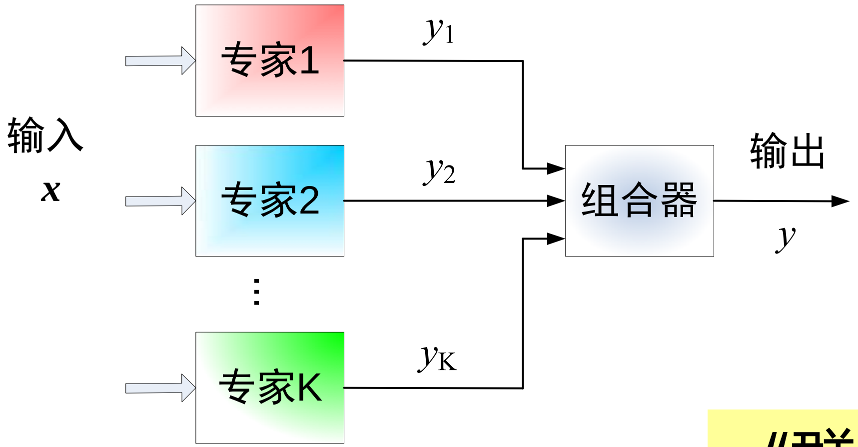
$$E_{COM}=\frac{1}{K}E_{AV}\ll E_{AV}$$

总体平均法: 减小方差

[1] Breiman, L. (1996). Bagging predictors. Machine Learning 26, 123-140.

总体平均法的缺陷

- 不同专家的误差往往是**高度相关的** “英雄所见略同”
  - **比如: 使用自助数据集(即数据集的切片或子集)来训练的一组专家**



$$y = \frac{1}{K}\sum_{i=1}^{K} y_i$$

“群众的智慧”

$$\frac{1}{K}E_{AV} \le E_{COM} \le E_{AV}$$

- 例1: 等概生成2个Gauss分布的类别：10个专家(采用不同初始条件的MLP)
  - P_{AV}=79.37%, **P_{COM}=80.27%**, P_C=81.51%

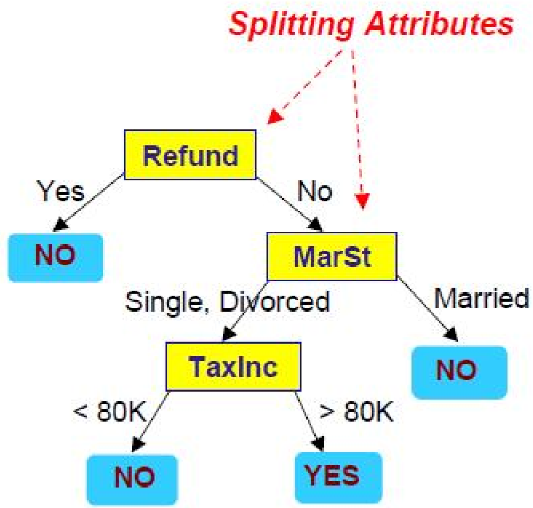
决策树(Decision Tree)

- 基于树结构进行决策
  - **分而治之——在决策过程中基于各个属性逐个提出判定问题**
  - **决策树的生成是个递归过程**
    - **关键问题：如何选择最优划分属性**

➤ 举例: 判断用户是否欺诈

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

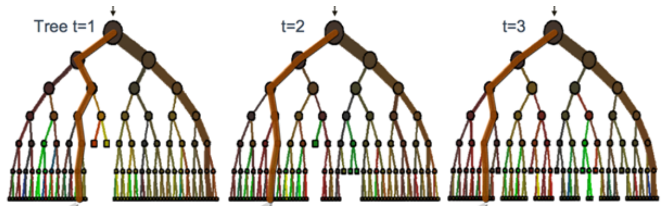
Training Data



Model: Decision Tree

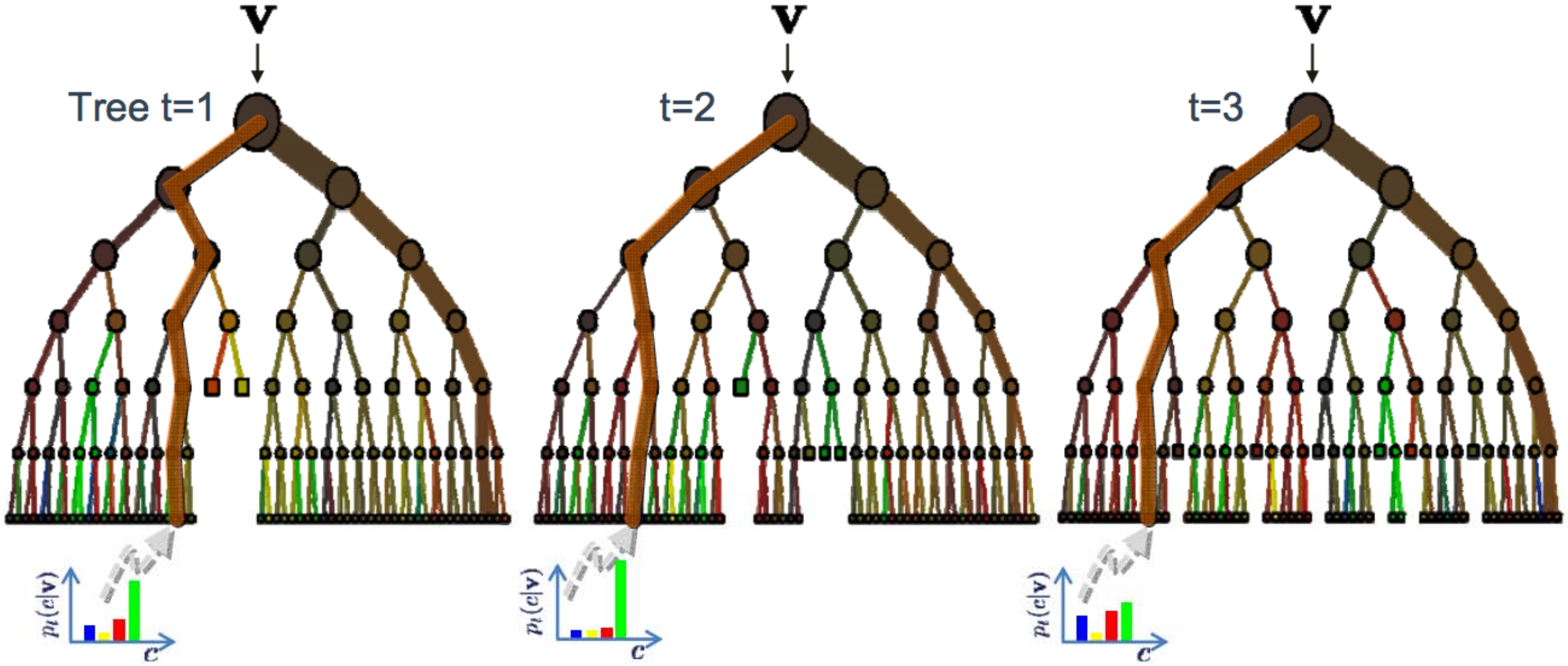
随机决策森林(Random Decision Forest)

- 决策树的集成(Ensemble of Decision Tree)
- 随机森林的构造算法：



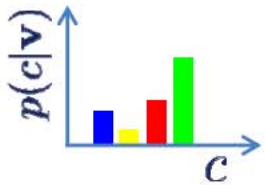
  – Step 1. 构造L棵树
    - 从训练数据集中随机抽取一个样本子集 j (j=1,...,L)
    - 生成森林里的第 j 棵树
      – (a) 从p维特征里随机选择 m 维 (m < p)
      – (b) 选择m个变量中的最佳分裂，比如采用信息增益法
      – (c) 分裂节点为子节点
  – Step 2. 对L棵决策树的输出采用简单平均法进行集成

例：随机决策森林



The ensemble model

Forest output probability $p(c|\mathbf{v}) = \frac{1}{T}\sum_{t}^{T} p_t(c|\mathbf{v})$



随机决策森林

- 应用：
  - – Object detection
  - – Kinect
  - – Image Classification

- 参考资料
  - Criminisi, Antonio; Shotton, Jamie; Konukoglu, Ender (2011). "Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning".

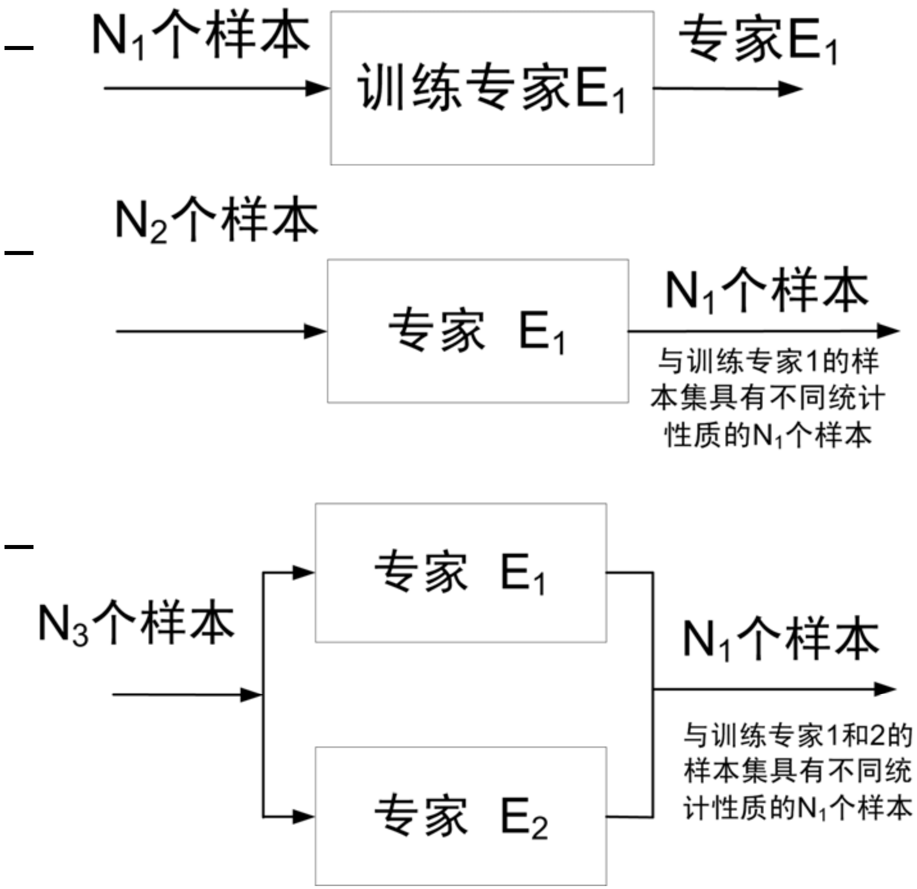
    Foundations and Trends in Computer Vision 7: 81–227.

专题 五：学习过程的统计性质与集成学习

- **内容提要**
  - 学习过程的统计性质
    - 误差分解
    - 偏倚/方差困境
  - 集成学习
    - 平均法
    - 推举法
    - 混合专家

图说过滤推举

- 设有3个专家



$$y = \frac{1}{3} \sum_{i=1}^{3} y_i$$

示例：分别用于3个专家的样本

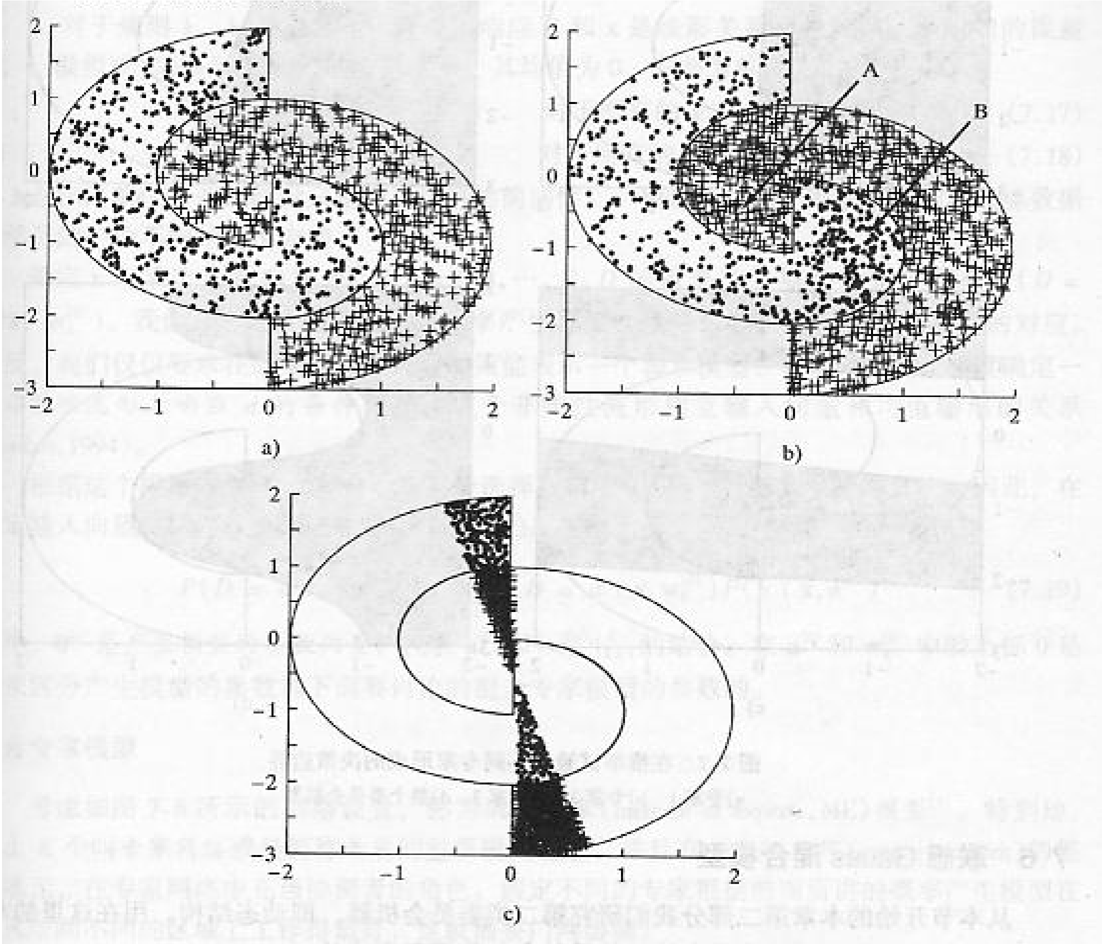
- 

图 7-6 推举的计算机试验中用于专家训练的样本散布图
a)专家 1 b)专家 2 c)专家 3

示例：不同专家形成的决策边界

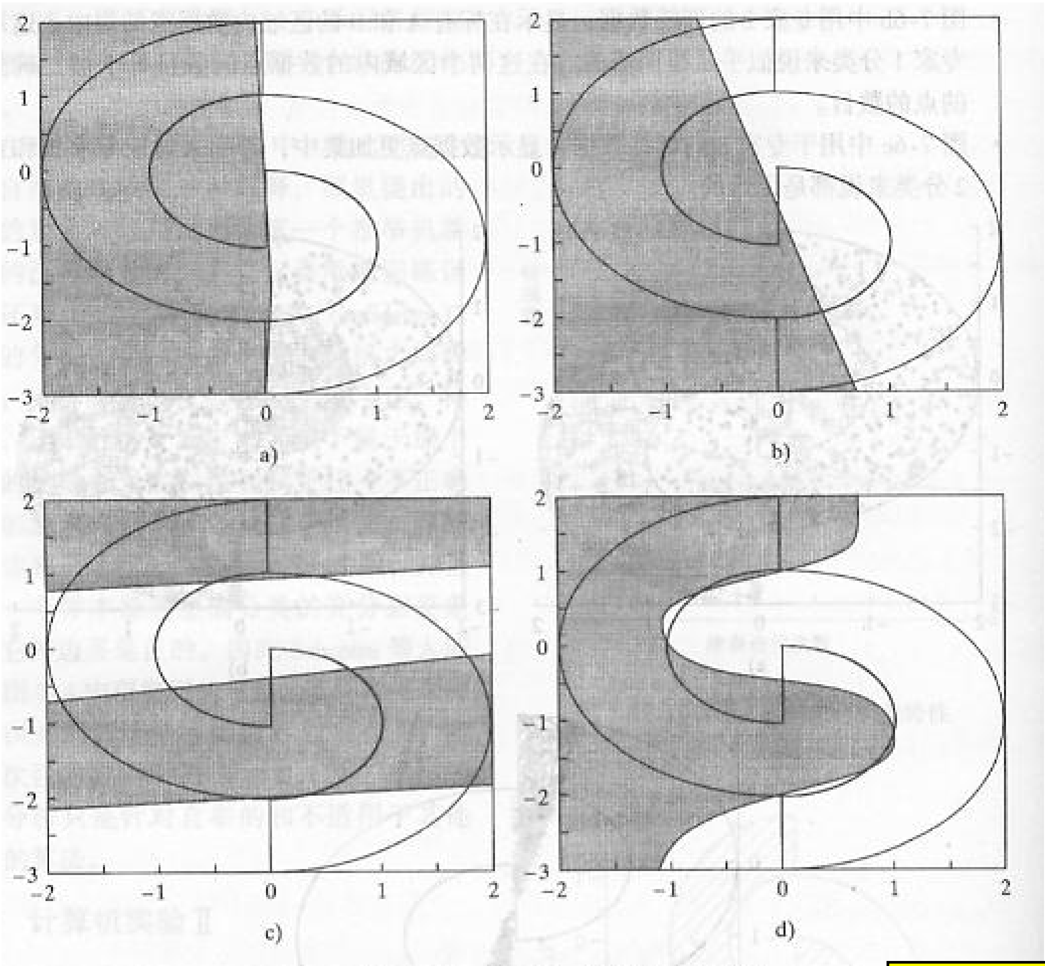
- 

图 7-7　在推举试验中不同专家形成的决策边界
a)专家 1　b)专家 2　c)专家 3　d)整个委员会机器

三个专家的识别正确率:

P_1=75.15%
P_2=71.44%
P_3=68.90%

基于过滤的推举法的委员会机器:

$$y = \frac{1}{3}\sum_{i=1}^{3} y_i$$

P_{Boost}=91.79%

“三个臭皮匠，顶个诸葛亮”

通过过滤推举

- 过滤的作用：
  - **专家E1的过滤和专家E1和E2的联合过滤，使得专家E2和专家E3能够集中于学习“难以学习”的部分**
- 说明：
  - **每个专家均利用N1个样本训练**
  - **共需要N1+N2+N3个样本**
    - **需要很大的训练样本集**

推举技术

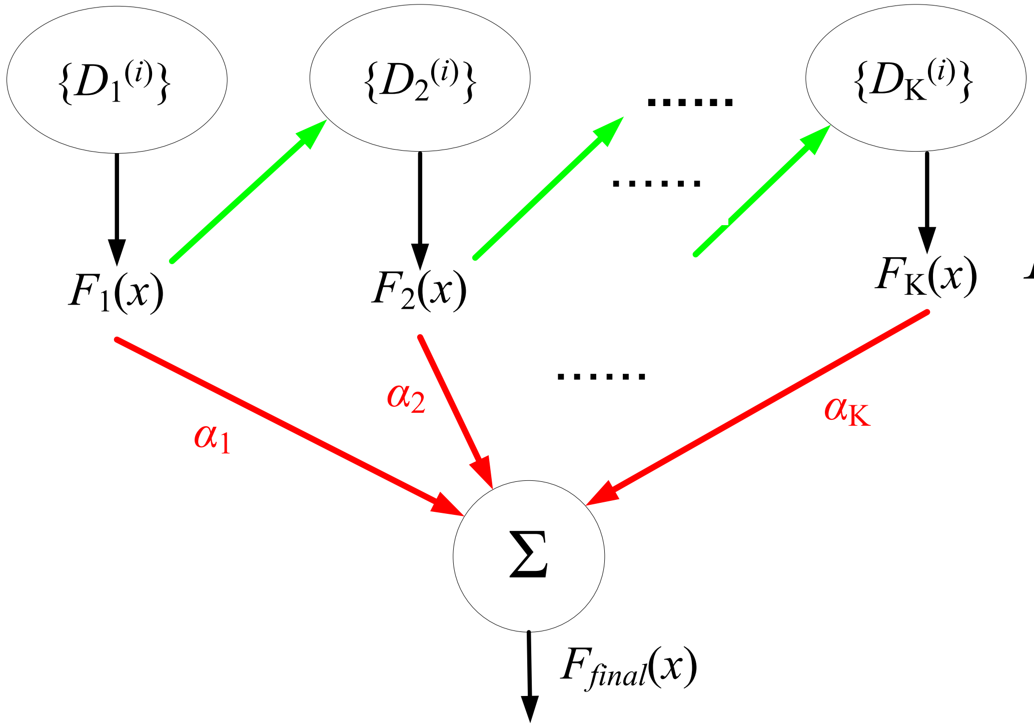
- 实现方法
  - **通过过滤推举**
  - **通过子抽样推举**
    - **自举 (Adaptive Boosting: AdaBoost)**
      - **通过重新加权推举**

通过子抽样推举：自举(AdaBoost)

- 动机
  - **克服通过过滤推举的需要大量样本的局限性**
- 特点
  - **允许训练数据重用**
- 基本思想
  - **训练样本集中能够被先前的弱学习模型正确分类的“容易”的样本被算法赋予较低的权值，而被经常错误分类的“难”的样本将被赋予较高的权值；**

图说AdaBoost

- 生成数据集的K个不同分布 $\left\{D_k^{(i)}\right\}_{i=1}^N$: $\left\{(\mathbf{x}_i, y_i), D_k^{(i)}\right\}_{i=1}^N, k=1,\dots,K$



$$F_{\text{final}}(\mathbf{x}) = \text{sign}\left(\sum_{k=1}^{K} \alpha_k F_k(\mathbf{x})\right)$$

其中,

$$\alpha_k = \frac{1}{2}\ln\frac{1-\varepsilon_k}{\varepsilon_k}$$

$$\varepsilon_k = \sum_{i=1}^{N} \mathrm{I}\left\{F_k(\mathbf{x}_i) \neq y_i\right\} D_k(i)$$

"通过前仆后继的努力，达到卓越"

AdaBoost的基本步骤

K: 学习模型数目；N是样本数

- 对于迭代k：k=1,…,K
  - **推举算法提供在训练样本X上分布D_k的弱学习模型**
    - **设弱学习模型**$F_k: X \rightarrow Y$**，它能正确分类训练样本的一 部分**
    - **初始分布：$D_1(i)=1/N$，i=1,…,N**
    - **更新方式：**

$$D_{k+1}(i)=\frac{D_k(i)}{Z_k}\exp\left(-\alpha_k y_i F_k(\mathbf{x}_i)\right)$$

其中Z_k是归一化常数

分布D_k如何计算？

  - **误差通过分布D_k度量**
    - **误差的计算：**

$$\varepsilon_k=\sum_{i=1}^{N}\mathrm{I}\left\{F_k(\mathbf{x}_i)\neq y_i\right\}D_k(i)$$

$$\text{if}\quad \varepsilon_k>\frac{1}{2},\ \text{then stop; otherwise continue.}$$

  - **最后把弱学习模型F_1，F_2，…，F_K合并成一个最终的强学习模型F_{final}**
    - **根据对于各个弱学习模型F_1，F_2，…，F_K加权求和**
      - **计算公式：**

最终模型如何计算？

$$F_{\text{final}}(\mathbf{x})=\operatorname{sign}\left(\sum_{k=1}^{K}\alpha_k F_k(\mathbf{x})\right)$$

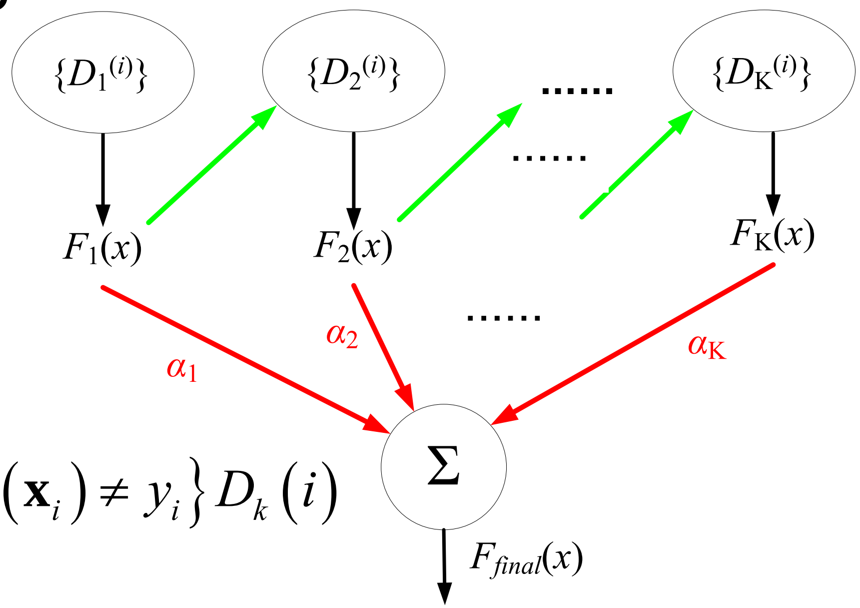
二值分类问题的AdaBoost算法

- 输入：
  - **训练样本{(x_i，y_i)}，N个标记样本的分布D**
  - **弱学习模型**F_1，F_2，…，F_K
- 初始化：
  - **对于i=1,…,N，设D_1**(i)=1/N
- 计算：对于k=1,…,K，进行如下过程
  - **1. 调用弱学习模型，对它提供分布D_k**
  - **2. 返回模型F_k: X→Y**
  - **3.计算模型F_k的误差：**

$$\varepsilon_k = \sum_{i=1}^{N} \mathrm{I}\left\{F_k\left(\mathbf{x}_i\right) \neq y_i\right\} D_k\left(i\right)$$

    - **If 误差大于0.5，停止；**
  - **4.更新分布D_k**

$$D_{k+1}\left(i\right) = \frac{D_k\left(i\right)}{Z_k} \exp\left(-\alpha_k y_i F_k\left(\mathbf{x}_i\right)\right)$$



- 输出: 最终模型

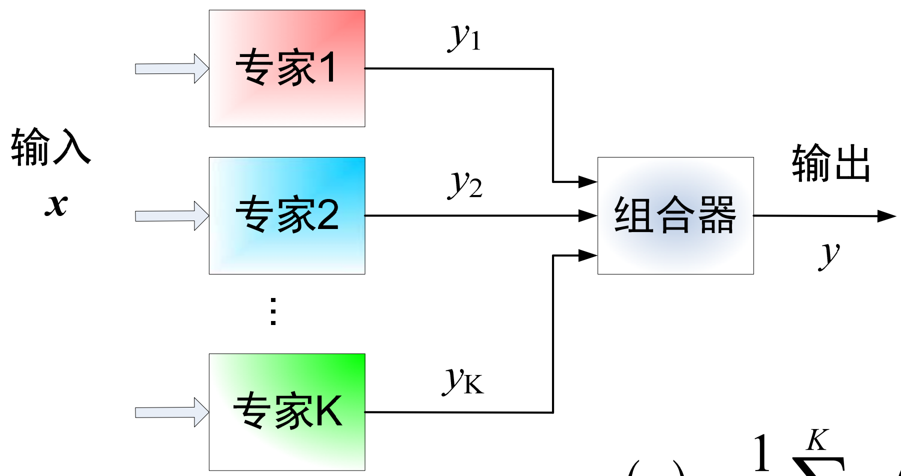
$$F_{\text{final}}\left(\mathbf{x}\right) = \text{sign}\left(\sum_{k=1}^{K} \alpha_k F_k\left(\mathbf{x}\right)\right),$$

集成学习的结构分类

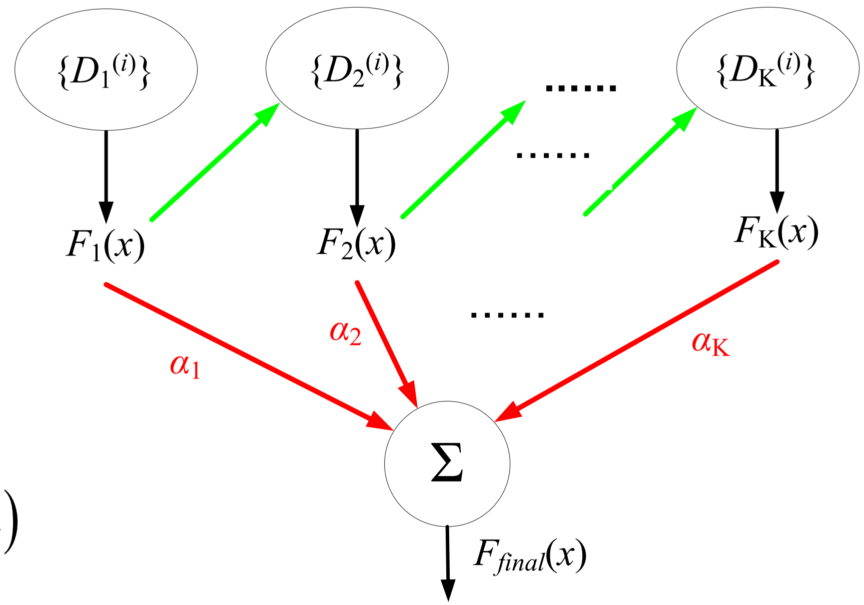
- 静态结构
  - 用于集成K个专家响应的机制与输入信号无关
    - 总体平均法
    - 推举(Boosting)方法
- 动态结构
  - 用于集成K个专家的响应的机制与输入信号相关

静态集成结构的缺陷

- 减小方差　　　　减小偏倚、减小方差



$$y(n)=\frac{1}{K}\sum_{i=1}^{K}y_i(n)$$



$$F_{final}(\mathbf{x})=sign\left(\sum_{k=1}^{K}\alpha_k F_k(\mathbf{x})\right)$$

“群众的智慧”　　　　**“前仆后继地不断努力”**

➢ 集成系数如何解释？比如专家A的权值是0.9…

专题 五：学习过程的统计性质与集成学习

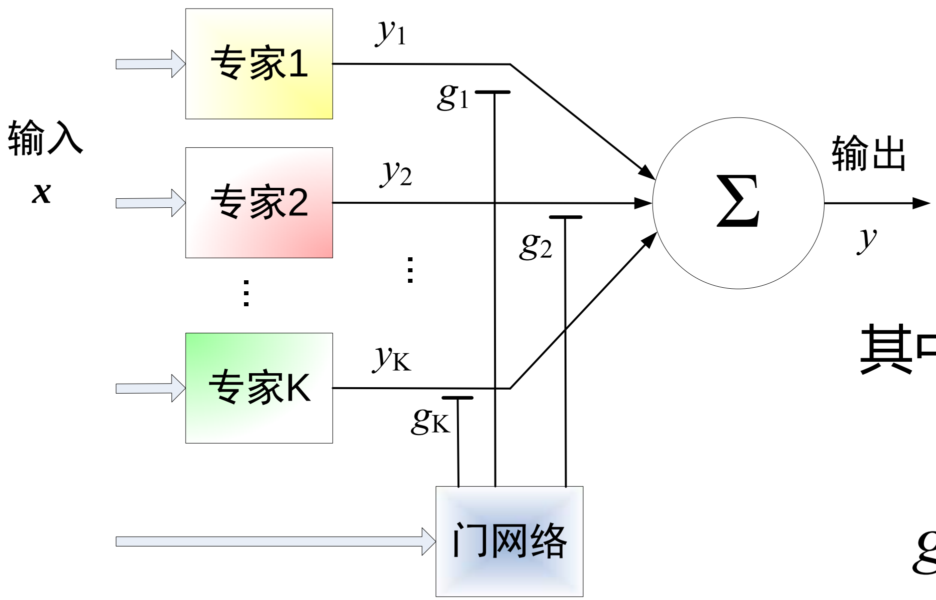
- **内容提要**
  - 学习过程的统计性质
    - 误差分解
    - 偏倚/方差困境
  - 集成学习
    - 平均法
    - 推举法
    - 混合专家

集成学习的结构分类

- 静态结构
  - 用于集成K个专家响应的机制与输入信号无关
    - 总体平均法
    - 推举(Boosting)方法
- 动态结构
  - 用于集成K个专家的响应的机制与输入信号有关
    - 混合专家模型：
      - 所有专家的单独响应通过单个门网络非线性地组合
    - 分层混合专家模型
      - 所有专家的单独响应通过多个门网络分层次地非线性组合

混合专家

- 整合专家知识的过程需要输入信号的参与



$$y=\sum_{k=1}^{K} g_k y_k=\sum_{k=1}^{K} g_k(\mathbf{x})\mathbf{w}_k^T\mathbf{x}$$

其中，$y_k=\mathbf{w}_k^T\mathbf{x}, k=1,2,\ldots,K,$

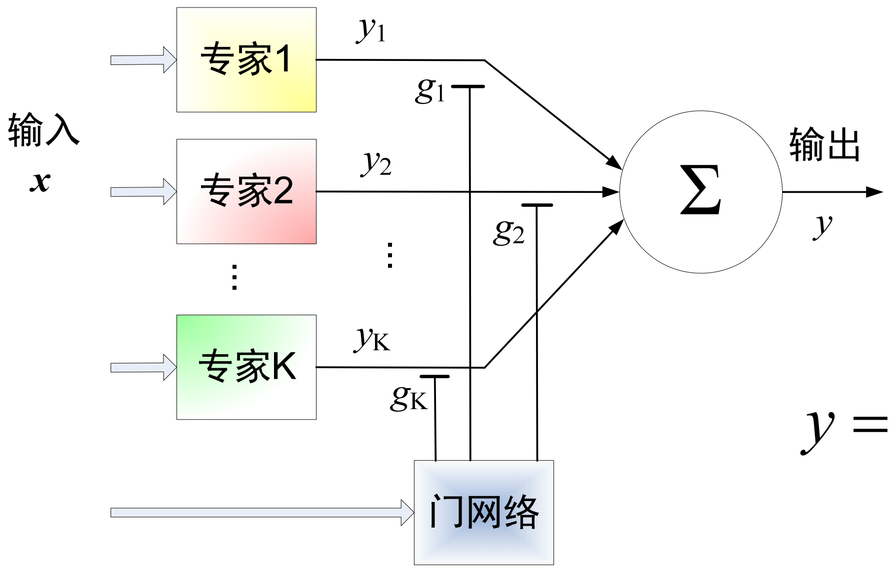
$$g_k(\mathbf{x})=\frac{\exp(u_k)}{\sum_{j=1}^{K}\exp(u_j)}=\frac{\exp(\mathbf{a}_k^T\mathbf{x})}{\sum_{j=1}^{K}\exp(\mathbf{a}_j^T\mathbf{x})}$$

软最大(softmax)

  - **ME (Mixture of Experts)模型**

ME vs. GMM

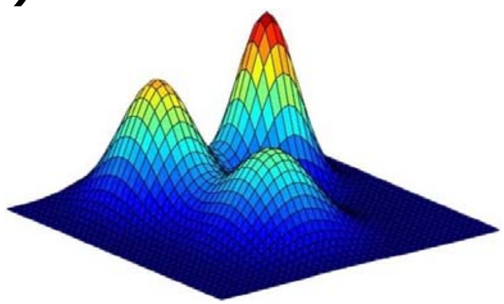
- ME



$$y = \sum_{k=1}^{K} g_k(\mathbf{x})\, y_k(\mathbf{x}, \mathbf{w}_k)$$

- GMM (Gaussian Mixture Model)

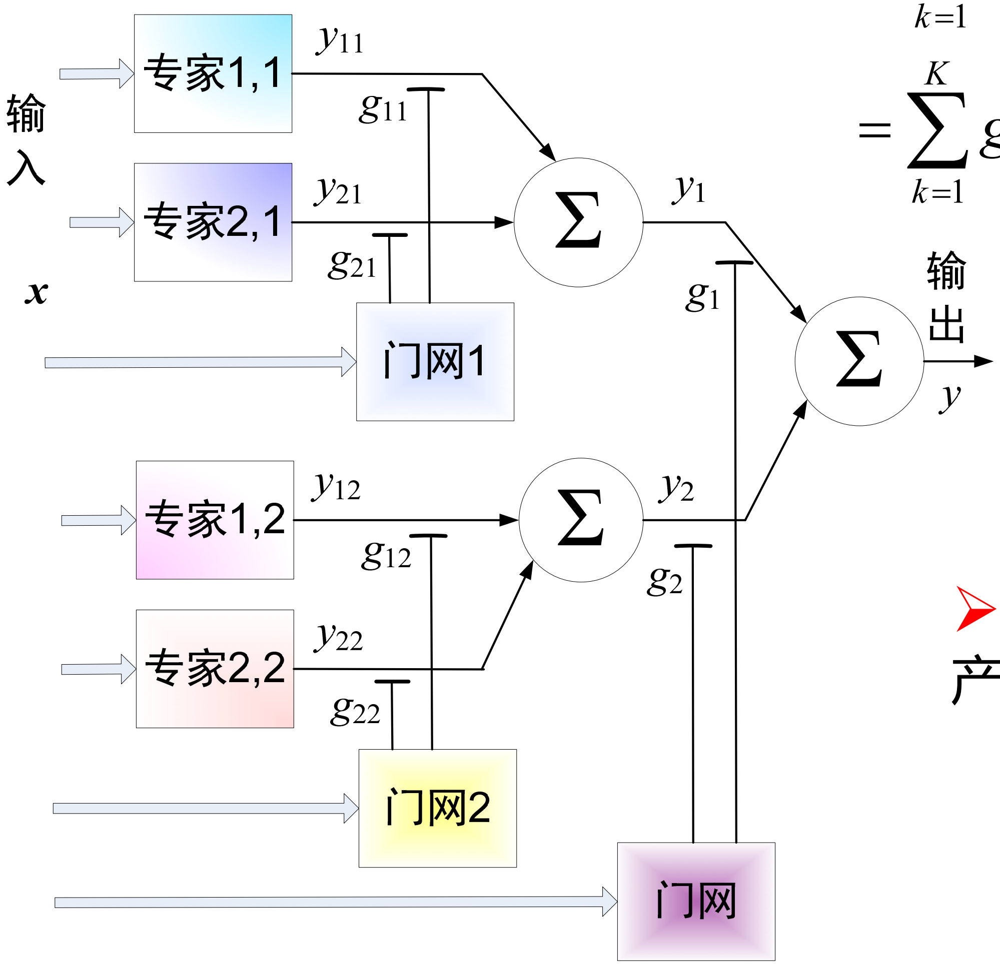
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k G(\mathbf{x} \mid \mu_k, \Sigma_k)$$



分层混合专家(HME)模型

- HME

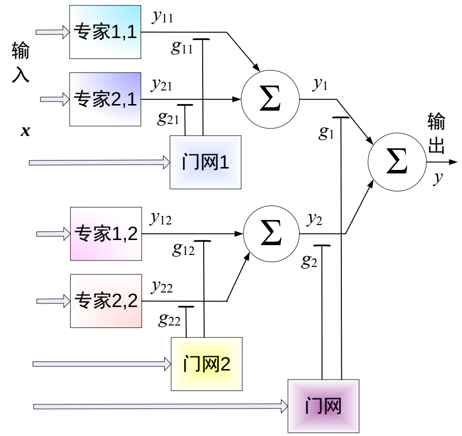
$$y=\sum_{k=1}^{K} g_k(x)\, y_k(x,\mathbf{w}_k)$$
$$=\sum_{k=1}^{K} g_k(x)\sum_{j=1}^{K} g_{j|k}(x)\, y_{j|k}(x,\mathbf{w}_{j|k})$$



➢ HME是分而治之策略的产物, 是软决策树

Jordan & Jacobs, "Hierarchical mixtures of experts and the EM algorithm", *Neural Computation*, 1994.

决策树 vs. HME



- 决策树

$$y(\mathbf{x})=\frac{1}{N(k)}\sum_{k=1}^{K}\mathrm{I}\{\mathbf{x}\in k\}\,y_k(\mathbf{x})$$

  – 对输入空间硬划分；在各划分内，只有一个模型产生响应

- 分层混合专家(HME)

$$y=\sum_{k=1}^{K}g_k(x)\,y_k(x,\mathbf{w}_k)=\sum_{k=1}^{K}g_k(x)\sum_{j=1}^{K}g_{j|k}(x)\,y_{j|k}(x,\mathbf{w}_{j|k})$$

其中，$g_k(x)=\dfrac{\exp(u_k)}{\sum_{j=1}^{K}\exp(u_j)}=\dfrac{\exp(a_k{}^T x)}{\sum_{j=1}^{K}\exp(a_j{}^T x)}$， $y_{j|k}(x)=\sum_{k=1}^{K}g_{j|k}y_{j|k}=\mathbf{w}^T_{j|k}\mathbf{x}$

$$g_{j|k}(x)=\frac{\exp(u_{j|k})}{\sum_{l=1}^{K}\exp(u_{l|k})}=\frac{\exp(a_{j|k}{}^T x)}{\sum_{l=1}^{K}\exp(a_{l|k}{}^T x)}$$

软决策树: 1. 对空间柔性划分; 2. 各个划分内所有模型均产生响应

HME的学习策略

- 最大似然估计：Likelihood

  1. **随机梯度方法**
     - **在线优化算法**

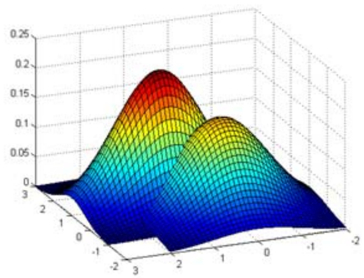
  2. **期望最大方法(EM: Expectation Maximization)**
     - **期望步骤(E):**
       - **使用非完整数据问题的观察数据集和参数向量的当前值，估计一个完整的数据集**
     - **最大化步骤(M)：**
       - **通过使E步产生的完整数据的对数似然函数的最大化来导出参数向量的一个新的估计值**

$$L_C(\Theta)=\prod_{i=1,\ldots N}\prod_{j=1,2}\prod_{k=1,2}\left[g_k^{(i)}g_{j|k}^{(i)}f_{jk}(d_i)\right]^{z_{jk}^{(i)}}$$

[1] Dempster, A. P., Laird N.M. and Rubin D.B. 1977, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Stasitstical Society., B, Vol.39, pp.1-38.
[2] Jordan & Jacobs, “Hierarchical mixtures of experts and the EM algorithm”, Neural Computation, 1994.

EM算法

- 基本思想
  - **从不完全或缺值(value-missing)数据出发，计算其极大似然估计**
    - **找到一组参数，使得非完整数据的对数似然函数取得最大**
- 导出途径
  - **变分推理**
    - **寻找目标函数的一个形式相对简单的下化该下界**
  - **借助缺失的或未观察到的变量**
    - **指示器(indicator)变量** z



N. Vasconcelos and A. Lippman, "Learning Mixture Hierarchies", NIPS, 1998

GMM的参数估计: EM算法

- 从未观测到变量角度理解EM算法
  - 设2个Gaussian分量

$$p(x)=(1-\pi)N(x;\mu_1,\sigma_1{}^2)+\pi N(x;\mu_2,\sigma_2{}^2)$$

引入隐含变量:

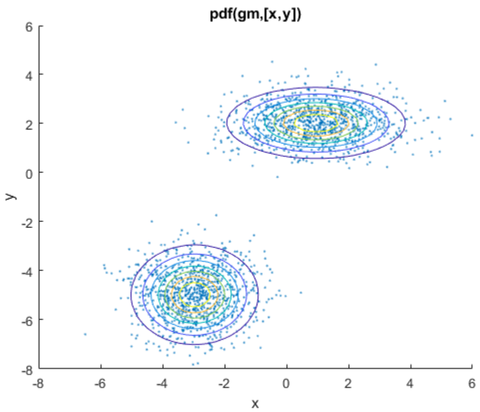
$$\Delta_i \in \{0,1\},$$

$$\pi = \mathrm{P}(\Delta_i = 1)$$

1. E-step ：计算响应度，对观测数据软指派

$$\gamma_i(\Theta)=E(\Delta_i \mid \Theta, Z)=\Pr(\Delta_i=1 \mid \Theta, Z)$$

$$\gamma_i=\frac{\pi N(x_i;\mu_2,\sigma_2{}^2)}{(1-\pi)N(x_i;\mu_1,\sigma_1{}^2)+\pi N(x_i;\mu_2,\sigma_2{}^2)}$$



2. M-step ：计算加权均值和协方差

$$\mu_1=\frac{\sum_{i=1}^{N}(1-\gamma_i)x_i}{\sum_{i=1}^{N}(1-\gamma_i)},\qquad \sigma_1{}^2=\frac{\sum_{i=1}^{N}(1-\gamma_i)(x_i-\mu_1)^2}{\sum_{i=1}^{N}(1-\gamma_i)},\qquad \pi=\frac{1}{N}\sum_{i=1}^{N}\gamma_i$$

Q / A

- Any Question?