

# 机器学习

# Machine learning

2011年春季学期

机器学习简介  
及  
决策树

刘扬  
哈工大计算机学院自然计算研究室

# 相关信息

- 教材

- Mitchell, **机器学习 (必需)**
- Chris Bishop, **Pattern Recognition and Machine Learning (电子版, 必需)**
- David Mackay, **Information Theory, Inference, and Learning Algorithms**

- 教师

- 刘扬
- [yliu76@hit.edu.cn](mailto:yliu76@hit.edu.cn)

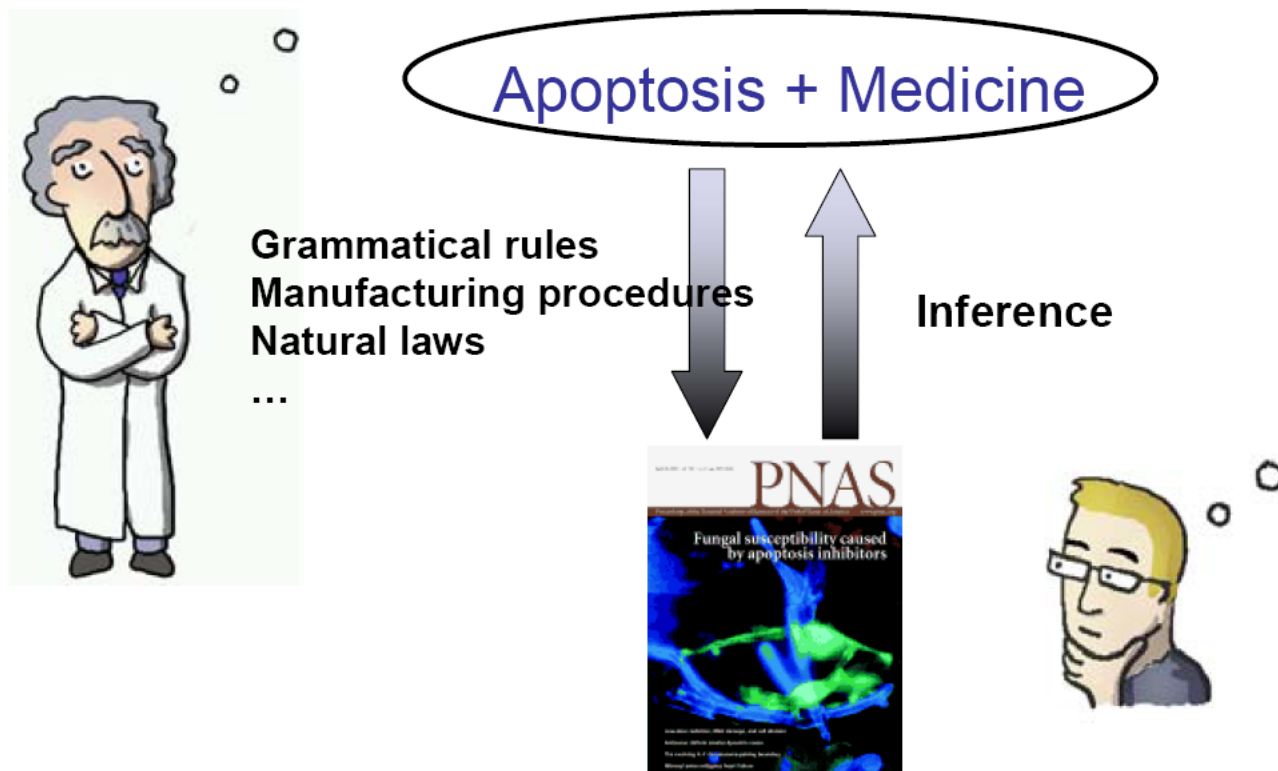
# 什么是学习？

- 定义1：由于经验或实践的结果而发生的持久或相对持久的适应性行为变化
- 定义2：能够使动物的行为对特定的环境条件发生适应性变化的所有过程，或者说是动物借助于个体生活经历和经验使自身的行为发生适应性变化的过程

○ 百度百科

# 什么是机器学习？

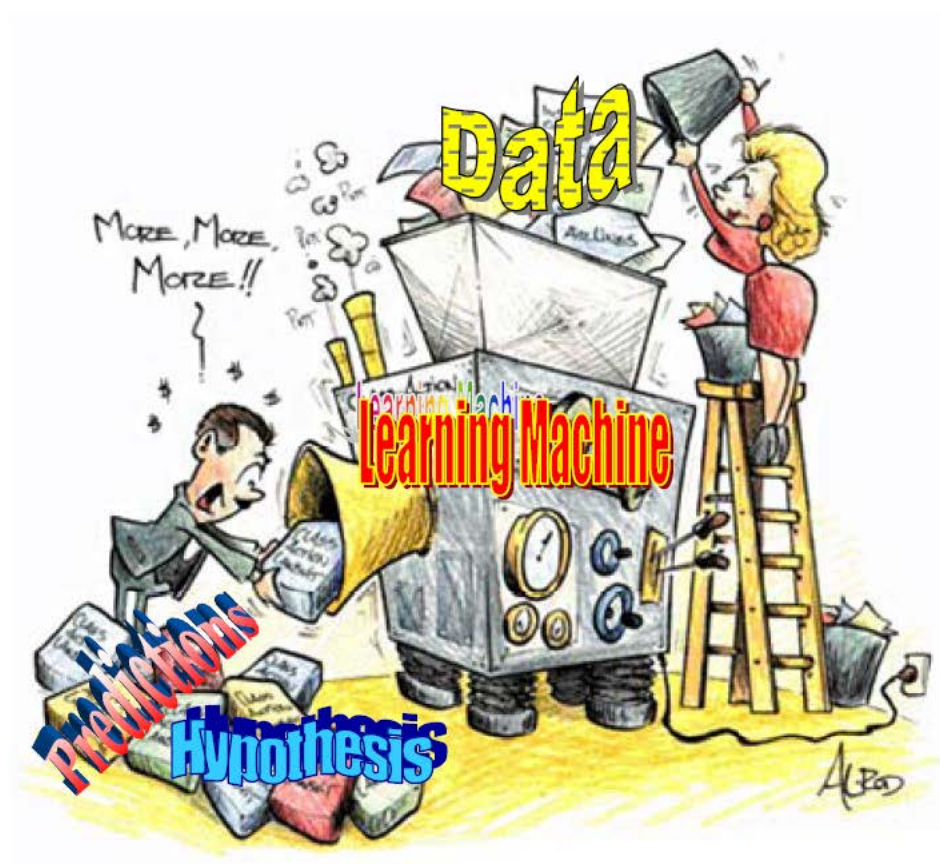
- 是寻找一种对自然/人工主题、现象或活动可预测且/或可执行的机器理解方法



# 什么是机器学习

- 研究计算机怎样**模拟或实现**人类（**动物**）的学习行为，以获取新的知识或技能
- 重新组织已有的知识结构使之不断改善自身的性能
- 是**人工智能**的核心，是使计算机具有智能的**重要途径**
- 其应用遍及人工智能的各个领域，它主要使用归纳、综合而不是演绎

# 机器学习的一个形象描述

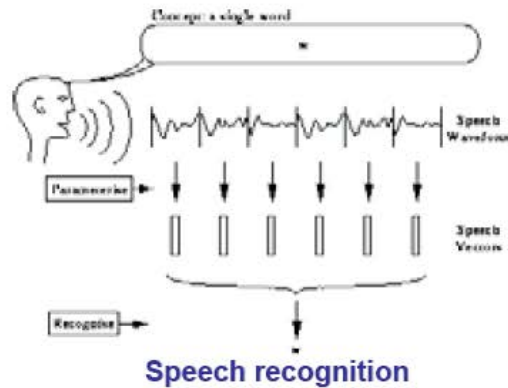


# 机器学习

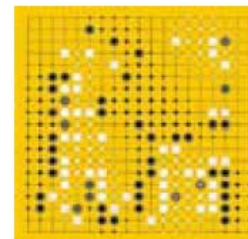
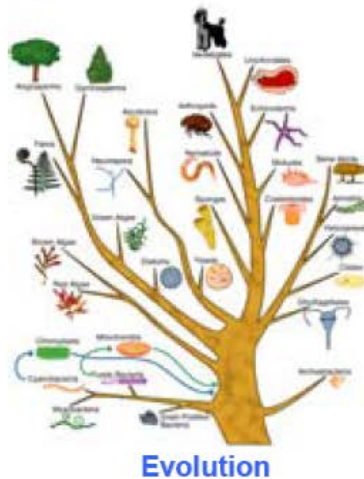
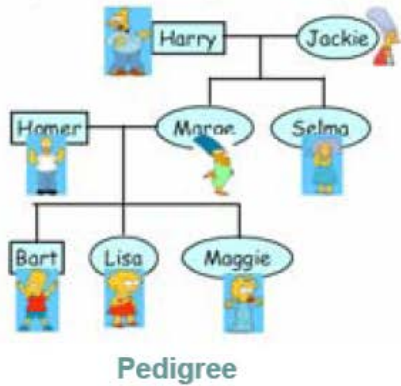
- 研究一种算法
  - 提高它的性能 ( P )
  - 在某项任务中 ( T )
  - 利用一些经验 ( E )

**well-defined learning task:  $\langle P, T, E \rangle$**

# 机器学习的应用



Computer vision



Games



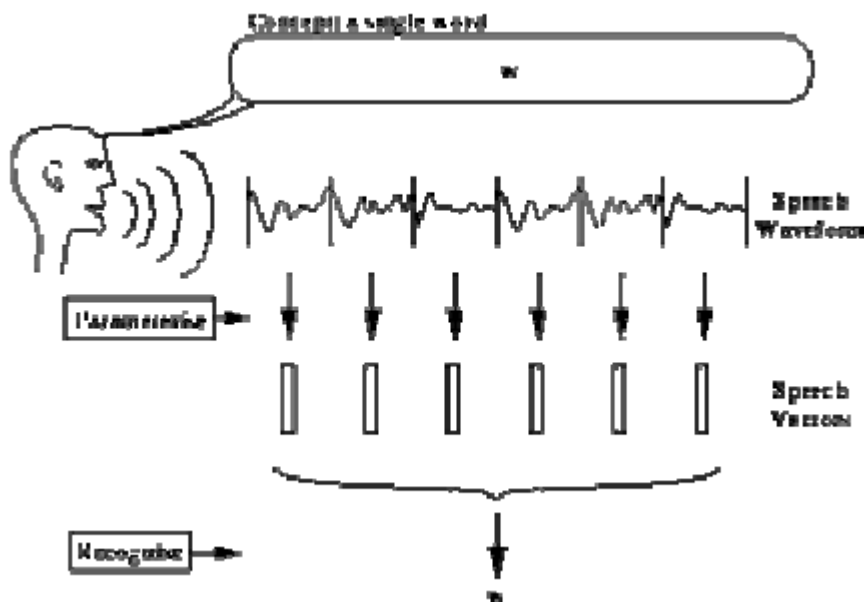
Robotic control





# 自然语言处理/语音识别

- 现在**语音识别器**或**翻译器**几乎都是建立在某种具有学习能力的设备上--- 使用的越多，则它越聪明



# 对象识别

- Behind a security camera, most likely there is a computer that is learning and/or checking!



# 机器人控制

- 汽车自动驾驶系统



# 机器人控制

- The **best** helicopter pilot is now a computer!
  - it runs a program that learns how to fly and make acrobatic maneuvers by itself!
  - no taped instructions, joysticks, or things like that ...



# 文本挖掘

- Reading, digesting, and categorizing a vast text database is too much for human!



- We want:



"Arts"	"Highlights"
NEW	MILLION
FILM	TAX
SHOW	PROGRAM
MUSIC	BUDGET
MOVIE	BILLION
PLAY	FEDERAL
MUSICAL	YEAR
BUFF	SPENDING
ACTOR	NEW
FIRST	STATE
YORK	PLAN
OPERA	MONEY
THEATER	PROGRAMS
ACTRESS	GOVERNMENT
LOVE	CONGRESS
	FAMILIES
	TEACHERS
	WORK
	PUBLIC
	TEACHER
	BENNETT
	MANGAT
	NAMPHY
	STATE
	PRESIDENT
	ELEMENTARY
	RAITI

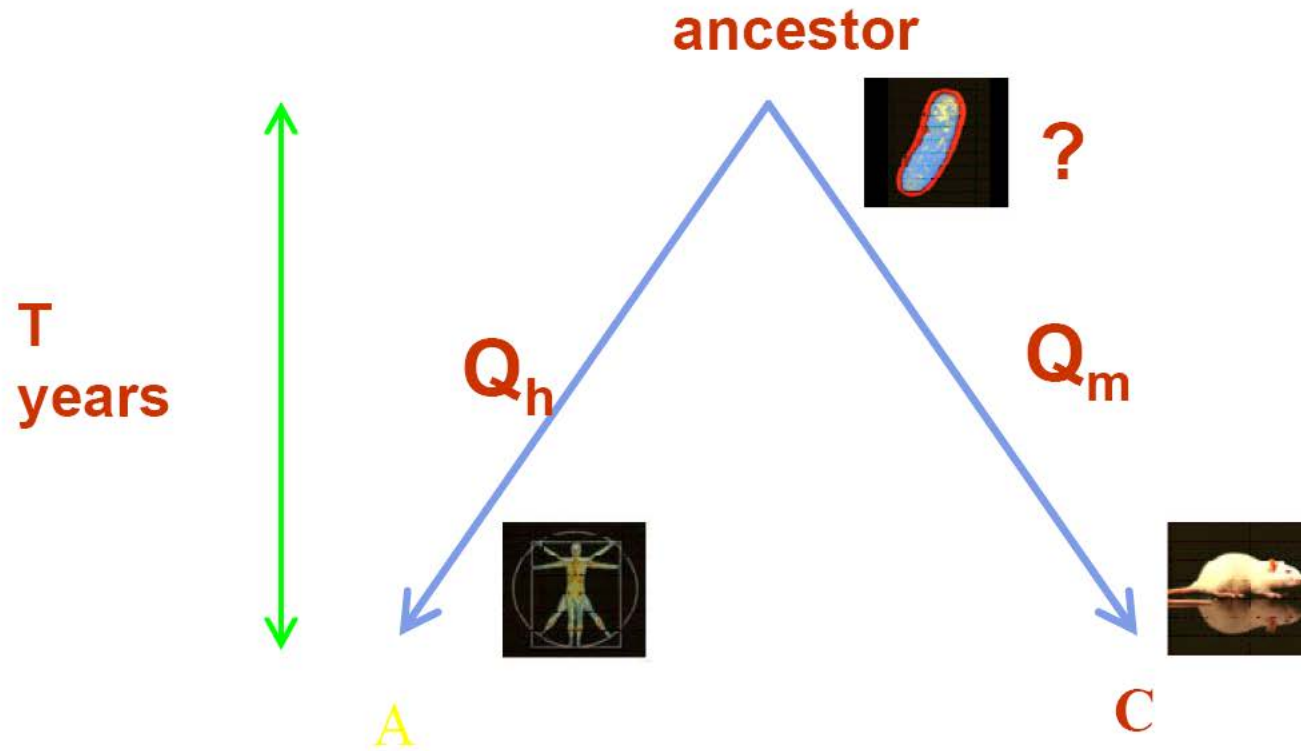
The Wilson Roadolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants as not every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Roadolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be \$300,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln



# 生物信息学

[illegible]

# 进化



# 机器学习的一般泛型

## ● 监督学习

Given  $D = \{\mathbf{X}_i, \mathbf{Y}_i\}$ , learn  $F(\cdot; \theta)$ , s.t.:  $\mathbf{Y}_i = F(\mathbf{X}_i)$   $D^{\text{new}} = \{\mathbf{X}_j\} \Rightarrow \{\mathbf{Y}_j\}$

## ● 无监督学习

Given  $D = \{\mathbf{X}_i\}$ , learn  $F(\cdot; \theta)$ , s.t.:  $\mathbf{Y}_i = F(\mathbf{X}_i)$   $D^{\text{new}} = \{\mathbf{X}_j\} \Rightarrow \{\mathbf{Y}_j\}$

## ● 强化学习

Given  $D = \{\text{env, actions, rewards, simulator/trace/real game}\}$

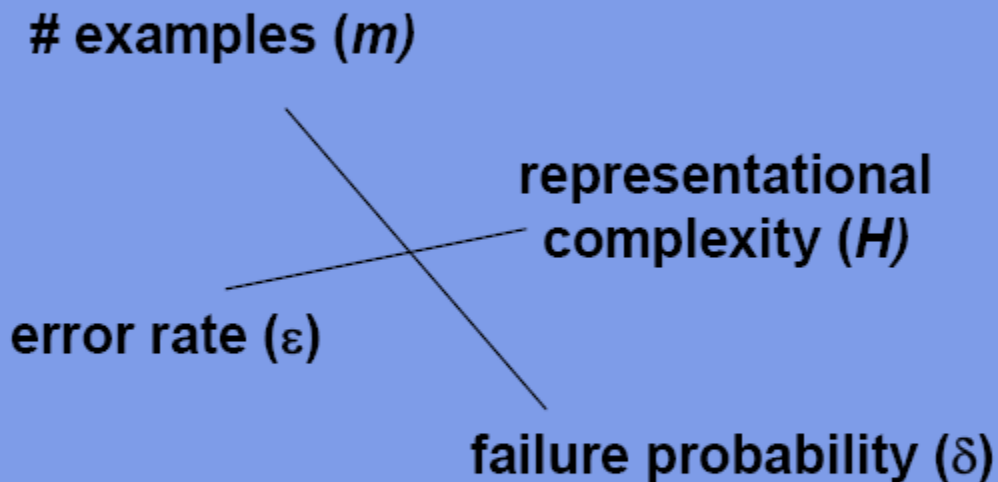
learn  $\text{policy} : e, r \rightarrow a$ , s.t.  $\{\text{env, new real game}\} \Rightarrow a_1, a_2, a_3 \dots$   
 $\text{utility} : a, e \rightarrow r$



# 机器学习

- 对于学习
- 一致性
- 偏执点
- 采样复杂度
- 学习率
- 收敛性
- 误差界
- 稳定性
- ...

## PAC Learning Theory (supervised concept learning)



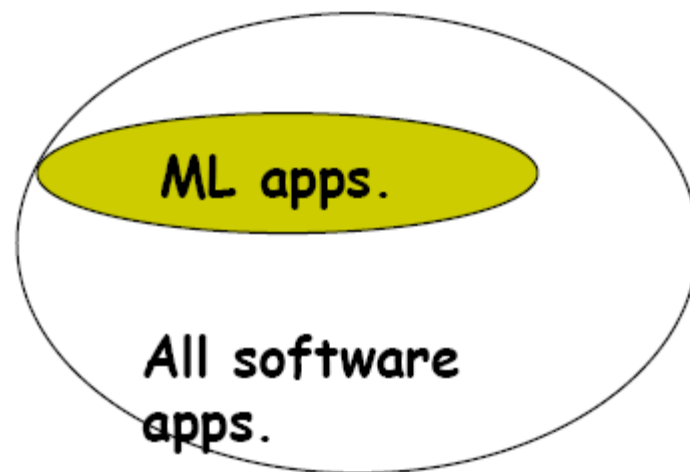
$$m \geq \frac{1}{\epsilon} (\ln |H| + \ln(1/\delta))$$

# 机器学习

- 机器学习是研究开发 一种具有如下能力的理论和计算机系统
  - 表示
  - 分类，聚类和识别
  - 不确定条件下推理
  - 预测
  - 对外界环境的反应
  - ...
- 可以在显示的模型或数学框架下，根据数据和自身经验，复杂的真实世界信息可以：
  - 被形式（formally）刻画和分析
  - 加入人的先验
  - 具有在数据与领域间的泛化和适应能力
  - 自动或自主操作
  - 被人类解释和感知

# 机器学习的快速增长

- 机器学习是最受青睐的方法
  - 语音识别，自然语言处理
  - 计算机视觉
  - 医学处理
  - 机器人控制
- 机器学习分量在增加
  - 机器学习性能不断提高
  - 数据快速增长，网络发展
  - 一些软件太复杂人工撰写比较困难
  - 新的感知器件/IO设备
  - 环境与用户的个性化服务



- 
- 推理 ( Inference )
  - 预测 ( Prediction )
  - 不确定性环境下的决策

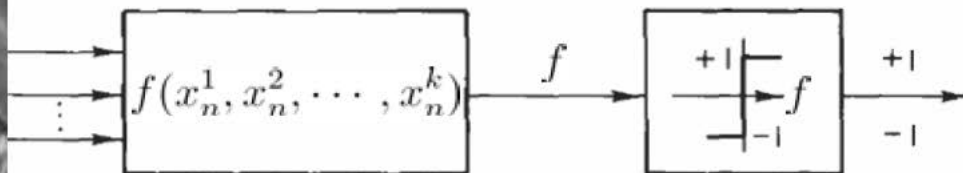
....

→ **统计机器学习**

→ **函数近似** :  $F(|\theta)$

# 分类

- 镰刀形细胞贫血症



- 函数近似

# 函数近似

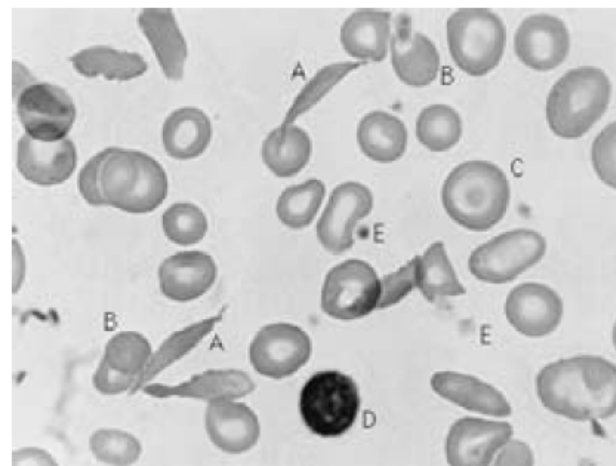
- 设置

- 示例集合  $X$
- 未知的目标函数  $f: X \rightarrow Y$
- 函数假设集合  $H = \{h|h: X \rightarrow Y\}$

- 给定

- 目标函数  $f$  的训练样本  $\{< x_i, y_i >\}$

- 确定  $h \in H$  , 可以最好地近似  $f$



# 税务欺诈检测问题

## Query Data

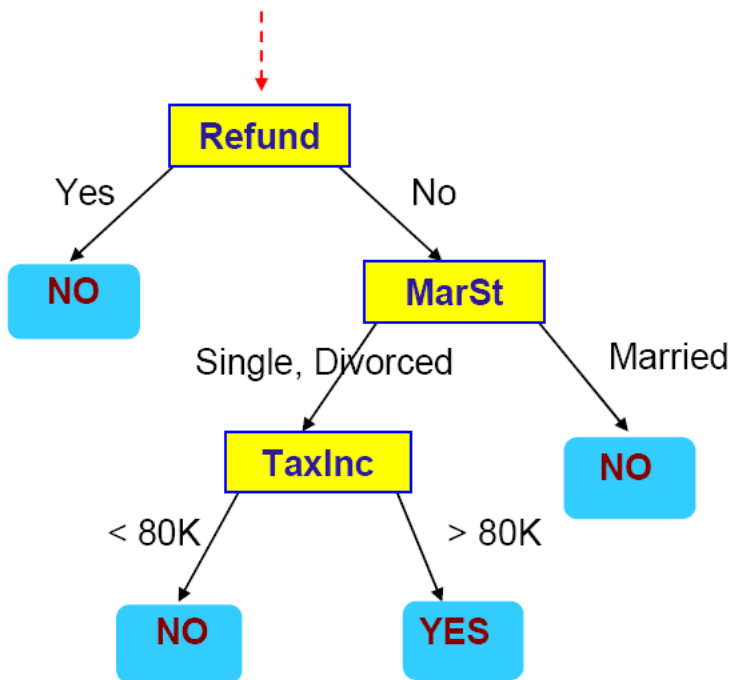
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

- 采用什么样的函数f
  - 假设空间是什么样的
- 如何使用？

# 查询数据的判别举例

- 从根节点开始

Start from the root of tree.



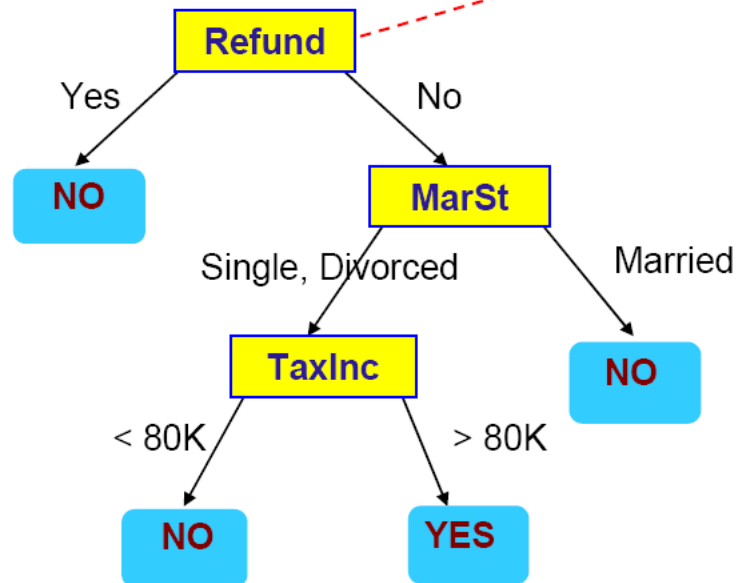
## Query Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



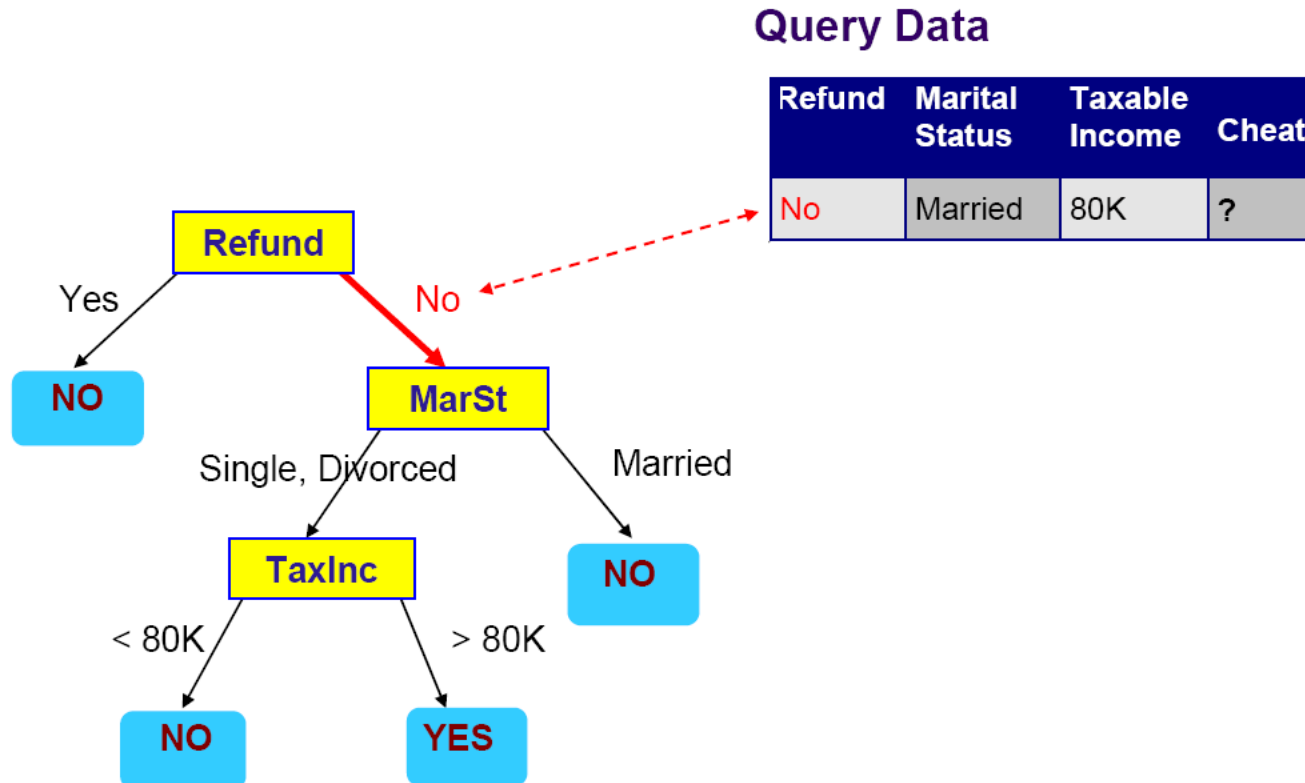
# 查询数据的判别举例

Query Data

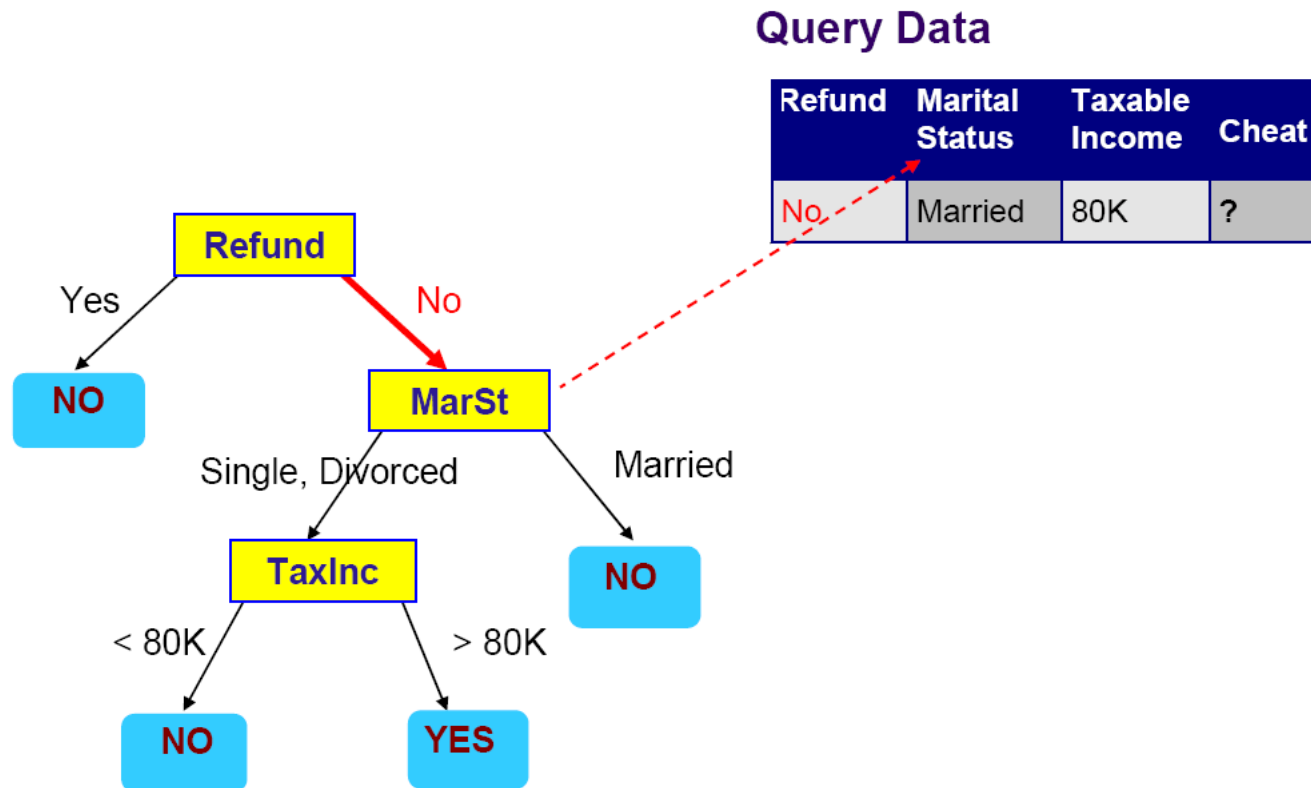


Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

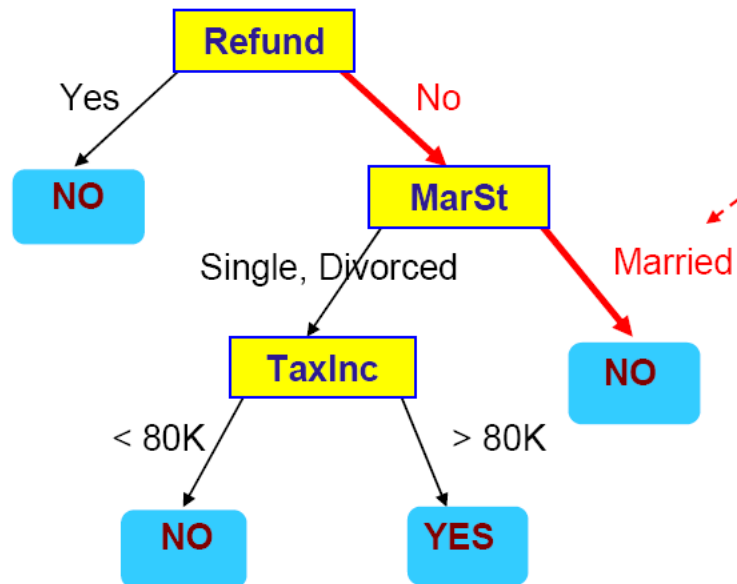
# 查询数据的判别举例



# 查询数据的判别举例



# 查询数据的判别举例



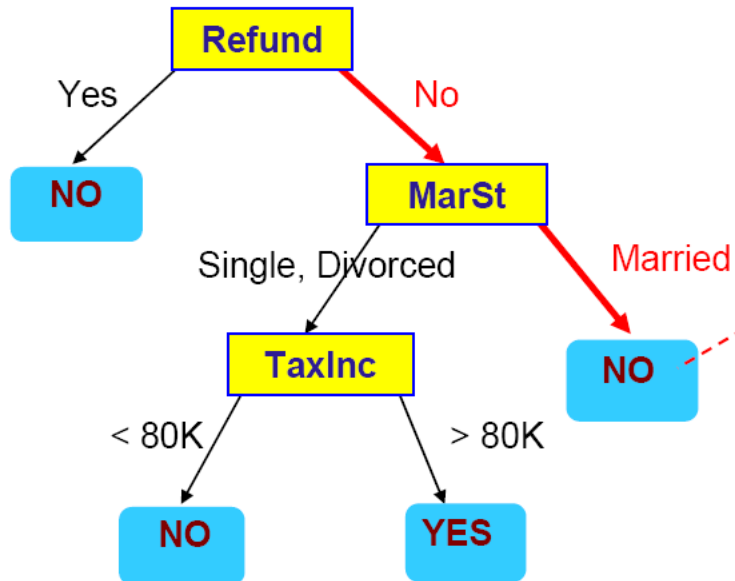
Query Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

# 查询数据的判别举例

## Query Data

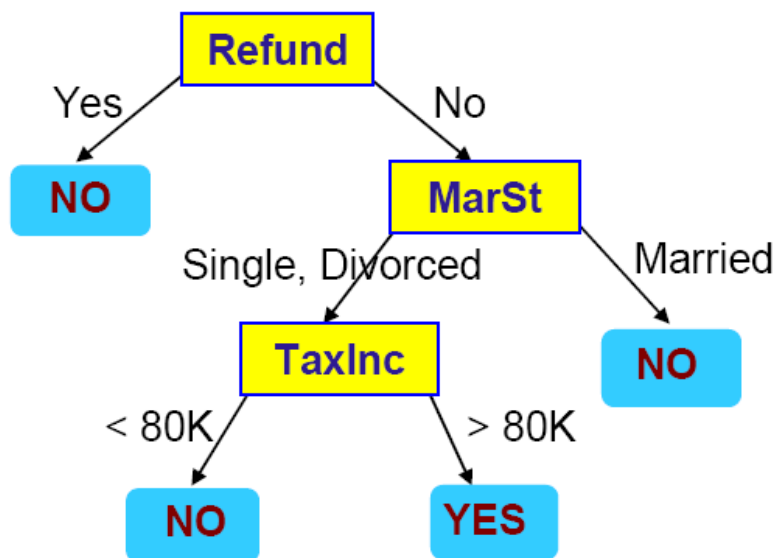
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"

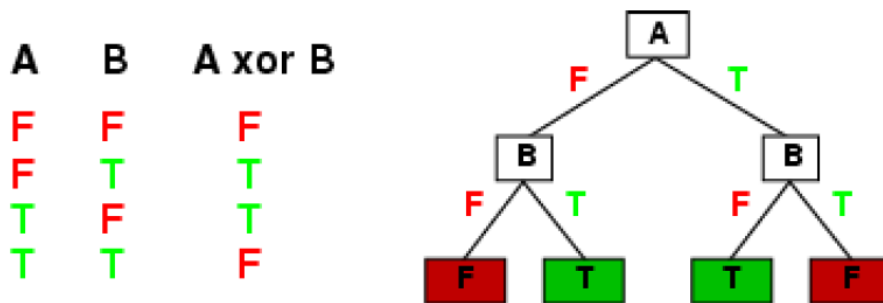
# 税务欺诈的一个假设（决策树）

- 输入：属性向量
  - $X=[\text{Refund}, \text{MarSt}, \text{TaxInc}]$
- 输出：
  - $Y$ =是否欺诈
- $H$  就是不同的决策过程（不同的决策树）
- 每一个内结点：测试一个属性 $X_i$
- 每个分支：选择属性 $X_i$ 的一个取值
- 每个叶结点：预测 $Y$



# 决策树的表示能力

- 决策树可以表示输入属性的任何函数
- 例如，对于布尔函数，真值表的一行→一条路径（根到叶）

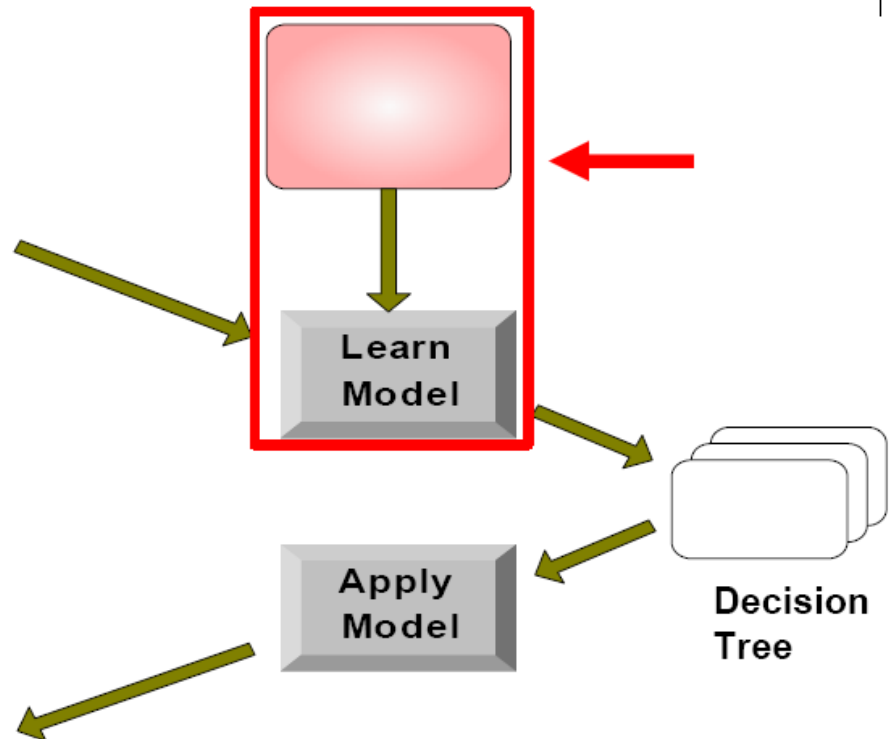


- 如果对训练数据中的每个样例都建立一条从根到叶的路径（除非对于输入x是不确定的）就得到一个一致的决策树，但其可能没有泛化能力
- 因此希望找一个更加紧凑（小规模）的决策树

# 决策树学习

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?



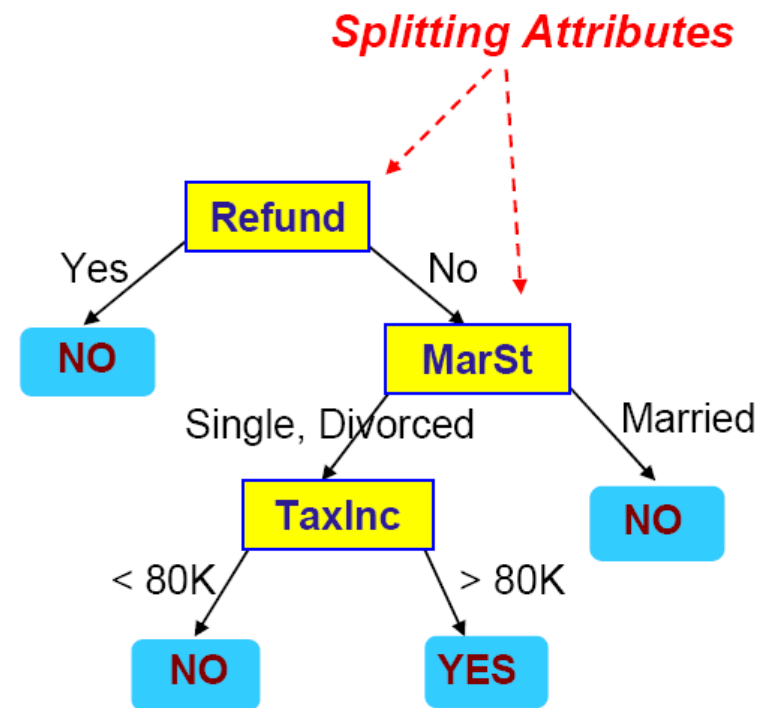


# 一棵决策树（欺诈问题）

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

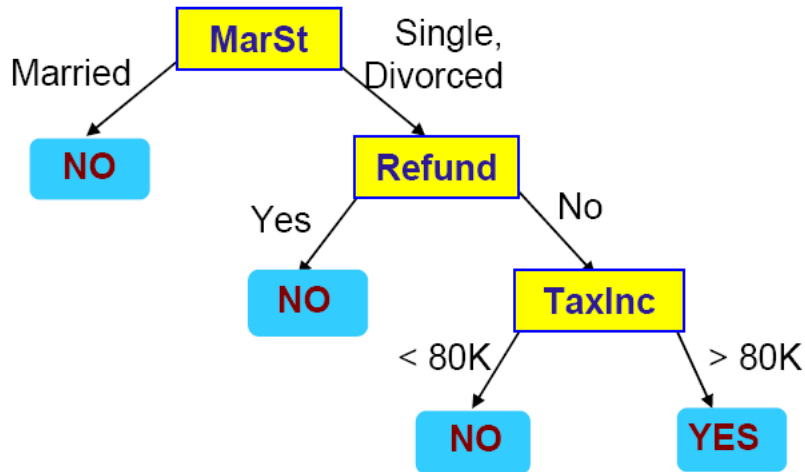


Model: Decision Tree

# 另一棵决策树

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

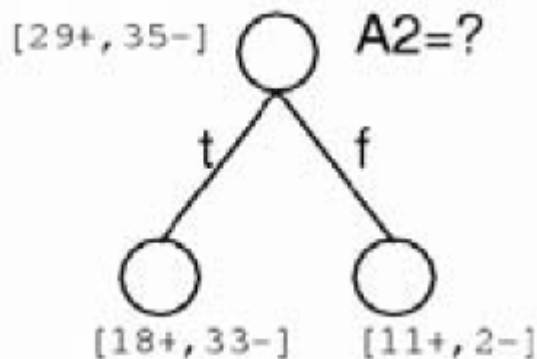
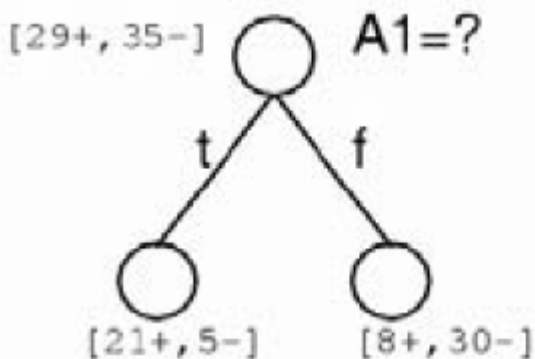
- 同样一个训练数据集，可以有多棵树与其一致

# Top-Down的决策树归纳算法（构造）

- Main loop :

1.  $A \leftarrow$  下一个结点  $node$  的最好属性
2. 把  $A$  作为决策属性赋给结点  $node$
3. 对  $A$  的每一个取值，创建一个新的儿子结点  $node$
4. 把相应的训练样本分到叶结点
5. 如果训练样本被很好的分类，则**停止**，否则在新的叶结点上重复上述过程

哪个属性更好？



# 树的归纳

- 贪心策略

- 基于一个可以最优化某项准则的属性来切分示例集和

- 问题

- 如何切分示例集和
  - 如何确定属性的测试条件？
  - 如何确定最好的切分？
- 停止切分准则

# 树的归纳

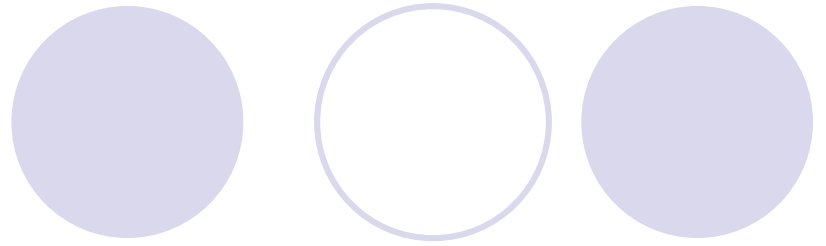
- 贪心策略

- 基于一个可以最优化某项准则的属性来切分示例集和

- 问题

- 如何切分示例集和
  - 如何确定最好的切分？
  - 如何确定属性的测试条件？
- 停止切分准则

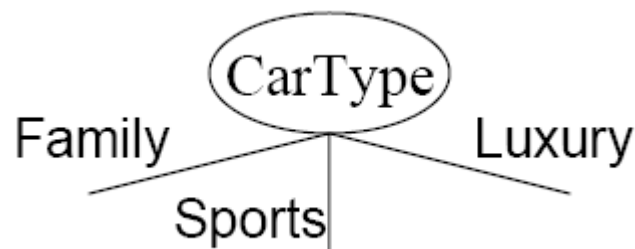
# 如何确定特测条件？



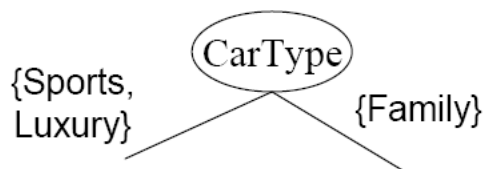
- 依赖于属性类型
  - 名词性\离散 ( Nominal )
  - 有序的 ( Ordinal )
  - 连续 ( Continuous )
- 依赖于切分的分支个数
  - 两路切分 ( 2-way )
  - 多路切分 ( Multi-way )

# 对名词属性的切分

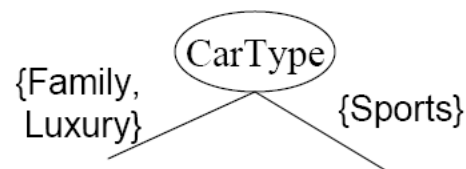
- 多路切分：一个离散属性值对应一路切分



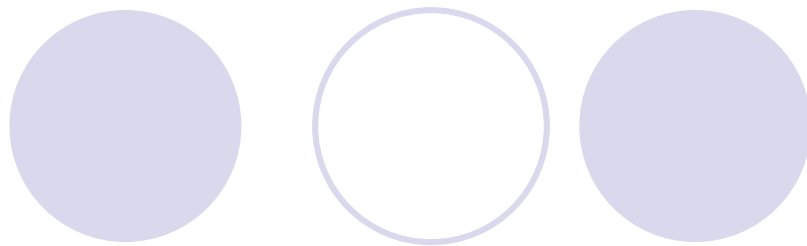
- 两路切分：离散属性值被切分成两个子集，  
需要寻找最优切分



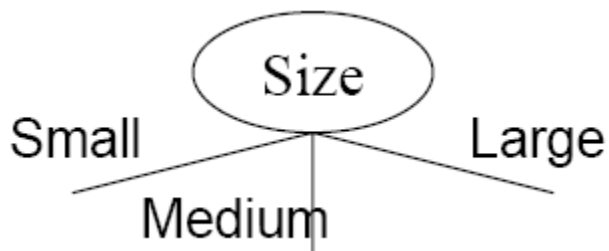
OR



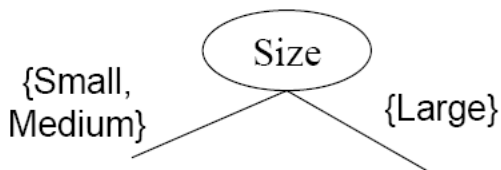
# 对有序属性的切分



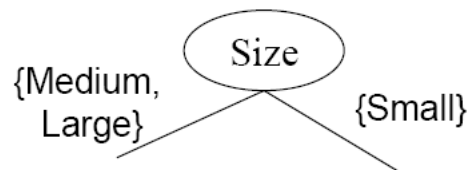
- 多路切分：一个属性值对应一路切分



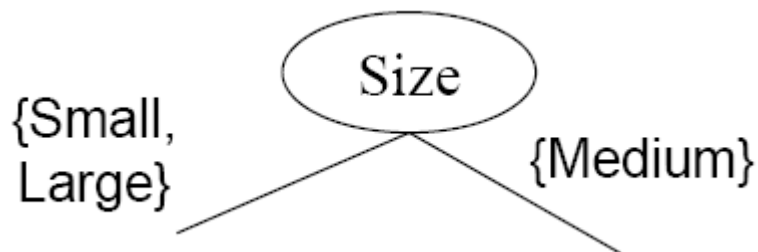
- 两路切分：属性值被切分成两个子集，**需要寻找最优切分**



OR

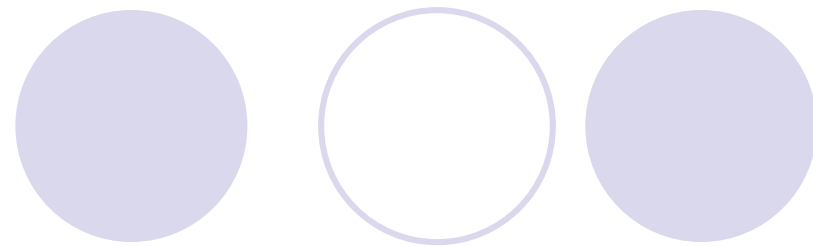


- 这个如何？**





# 对连续属性的切分



- 不同的处理方案

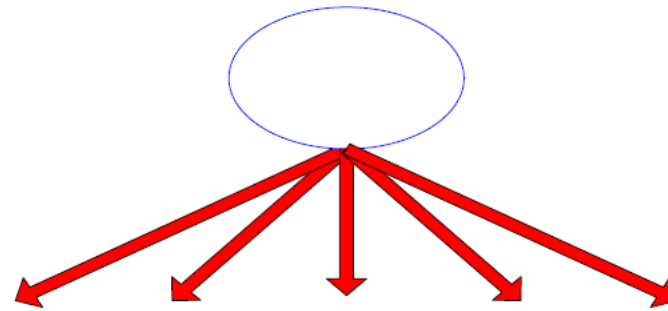
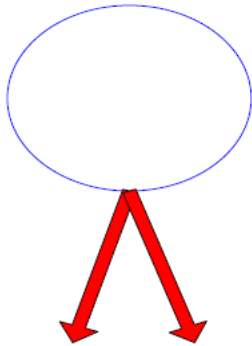
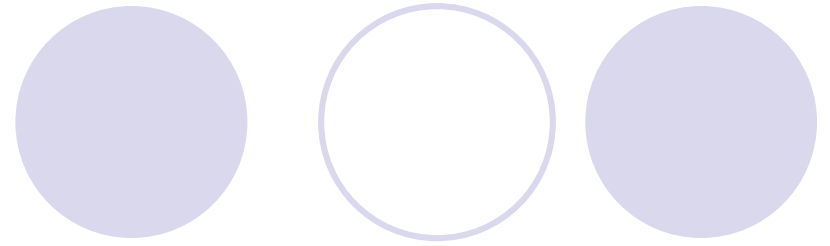
- 离散化构造有序的分类属性

- 静态——在起始位置一次离散化
    - 动态——范围可以通过等区间或等频率确定，或者是聚类

- 二值决策： $(A < V) \text{ or } (A \geq V)$

- 考虑所有可能的切分并选择最好的
    - 计算量可能非常大

# 对连续属性的切分



# 树的归纳

- 贪心策略

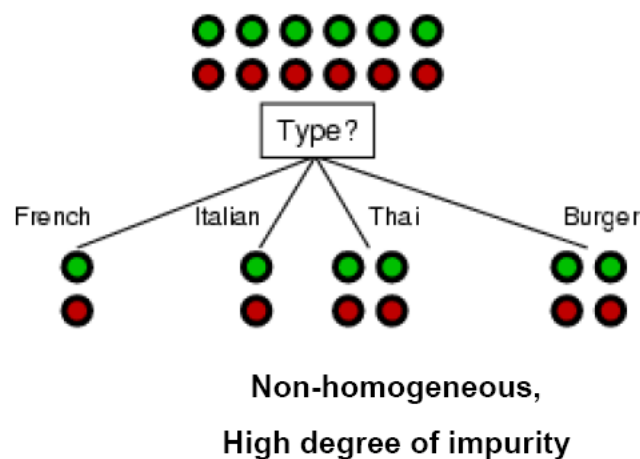
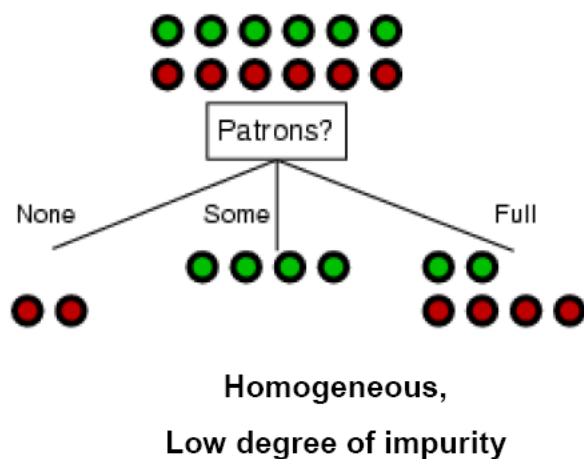
- 基于一个可以最优化某项准则的属性来切分示例集和

- 问题

- 如何切分示例集和
  - 如何确定最好的切分？
  - 如何确定属性的测试条件？
- 停止切分准则

# 如何确定最好的切分

- Idea：一个好的属性切分是将示例集合分成若干子集，**最理想情况是每个子集为“皆为正例”或“皆为反例”**



- 贪心搜索
  - 更倾向结点上的数据具有同质（homogeneous）类别分布
- 需要对结点混杂度（impurity）进行**测量**

# 如何衡量属性的“好”与“坏”

- 熵 ( Entropy )

- 随机变量 $X$ 的熵 $H(X)$

- $H(X)$ 是对从 $X$ 随机采样值在最短编码情况下的每个值平均 ( 期望 ) 长度 ( 以2为底就是0、1编码 )

$$H(X) = - \sum_{i=1}^N P(x = i) \log_2 P(x = i)$$

- Why ?

信息论：

在最短编码情况下，对消息 $X = i$ 分配 $-\log_2 P(X = i)$ 位，  
所以其编码一个随机变量 $X$ 的期望位数是

$$H(X) = - \sum_{i=1}^N P(x = i) \log_2 P(x = i)$$

# 如何衡量属性的“好”与“坏”

- 条件熵

- $X$ 在给定 $Y = v$ 特定条件熵 $H(X|Y = v)$

$$H(X|y = j) = - \sum_{i=1}^N P(x = i|y = j) \log_2 P(x = i|y = j)$$

- $X$ 在给定 $Y$ 条件熵 $H(X|Y)$

$$H(X|Y) = \sum_{j \in \text{Val}(y)} P(y = j) H(X|y = j)$$

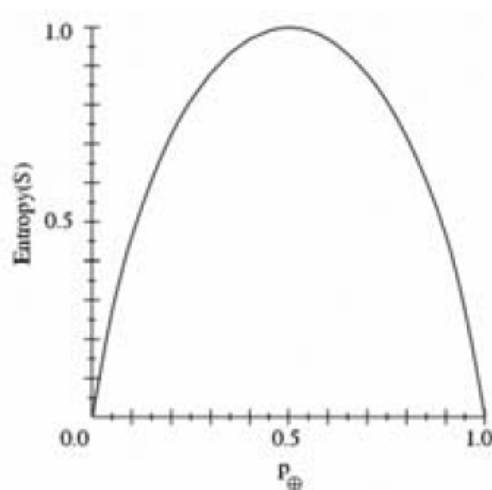
- $X$ 和 $Y$ 的互信息

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned}$$

# 样本熵

- S是训练例子的样本集
- $P_+$ 是S中的正例比例
- $P_-$ 是S中反例的比例
- 用熵来测量S的混杂度

$$H(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$



# 计算熵的例子

$$H(X) = - \sum_{i=1}^N P(x = i) \log_2 P(x = i)$$

C1	<b>0</b>
C2	<b>6</b>

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Entropy} = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

C1	<b>1</b>
C2	<b>5</b>

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Entropy} = - (1/6) \log_2 (1/6) - (5/6) \log_2 (1/6) = 0.65$$

C1	<b>2</b>
C2	<b>4</b>

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Entropy} = - (2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$$

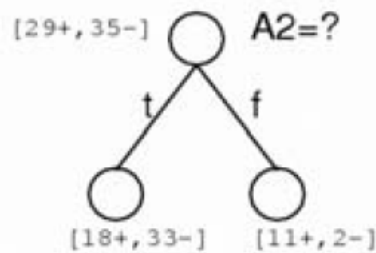
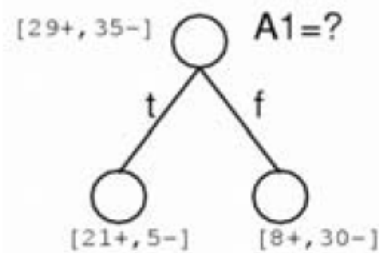


# 信息增益

- 信息增益

$$GAIN_{split} = Entropy(p) - \left( \sum_{i=1}^k \frac{n_i}{n} Entropy(i) \right)$$

- 父结点P被切分成k部分； $n_i$ 是每一切分的样本数  
即目标类变量与属性A变量在S（样本集）上的互信息



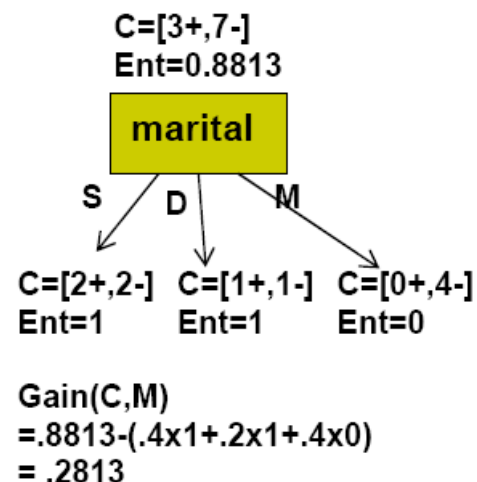
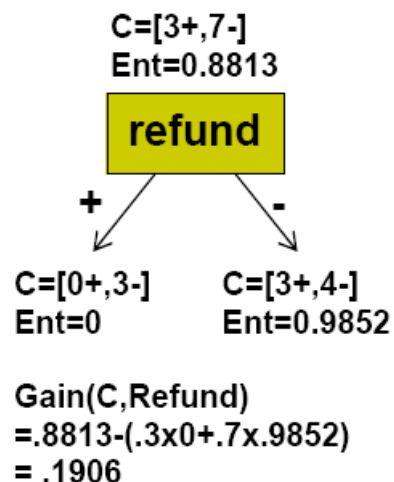
- 测量由于切分带来的熵减少量，选择具有最大减少量的切分（最大增益）
- 在ID3和C4.5中采用
- **缺点**：倾向选择具有切分分支多的属性，因为每份可以很少的样本，但很纯

# 例子

categorical  
categorical  
continuous  
class

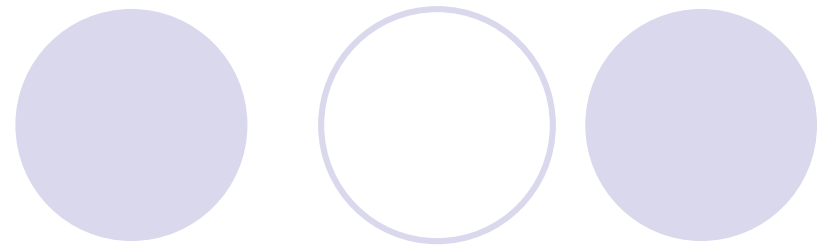
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



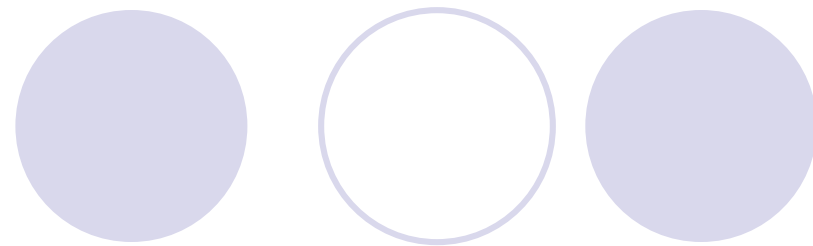
选择哪个属性作为根节点？

# 树归纳的停止准则



- 当一个结点上所有样本属于同一个类别，  
停止扩展
- 当一个结点上所有样本具有相似的属性值，  
停止扩展
- 提早结束（以后会介绍）

# 基于决策树的分类

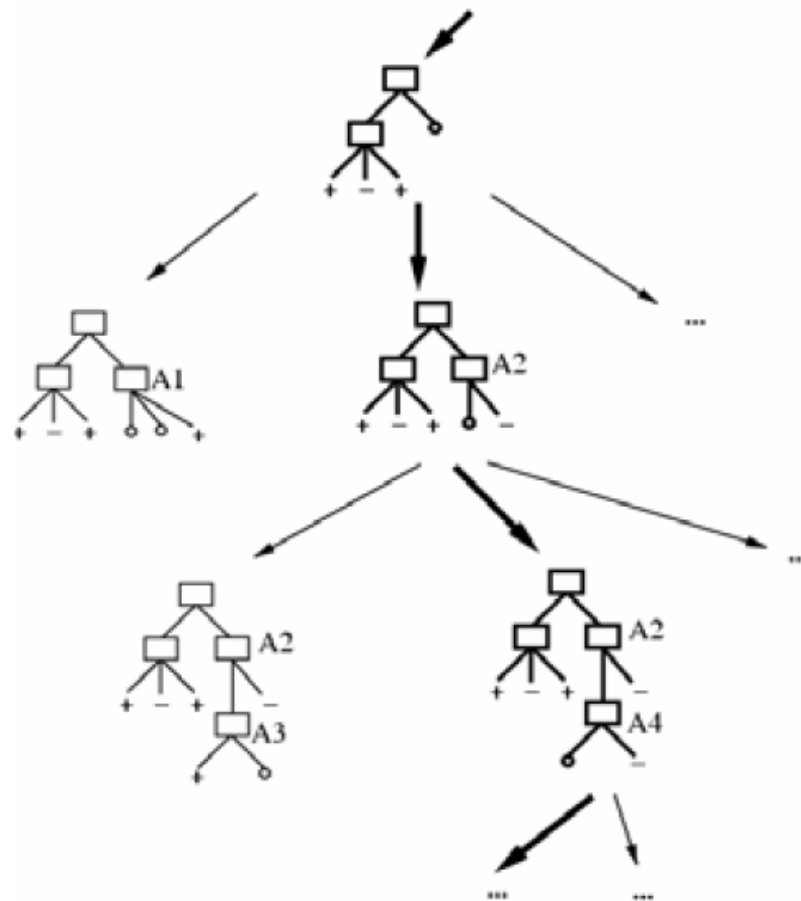


- 优点
  - 构建过程计算资源开销小
  - 分类未知样本速度极快
  - 对于小规模树比较容易解释
  - 在许多小的简单数据集上性能与其它方法相近
- 例子：C4.5
  - 深度优先构建方法
  - 采用了信息增益
  - 在每一个结点需要对示例依据连续属性排序
  - 数据需要全部装入内存
  - 不适合大规模数据

# 应该输出哪棵树

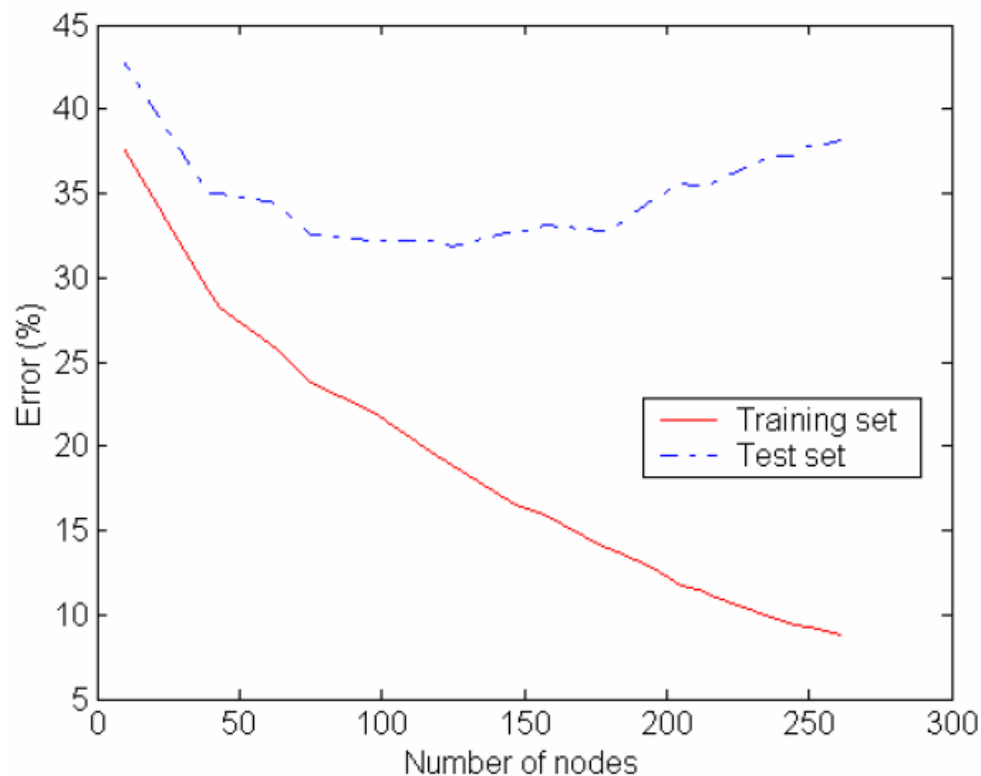
- ID3在决策树空间中执行启发式搜索
- 当获得**最小**可接受的决策树时则**停止**
- Why?

**Occam's 剃刀：选择适合训练集合数据的最简单假设**



# 决策树的实践问题

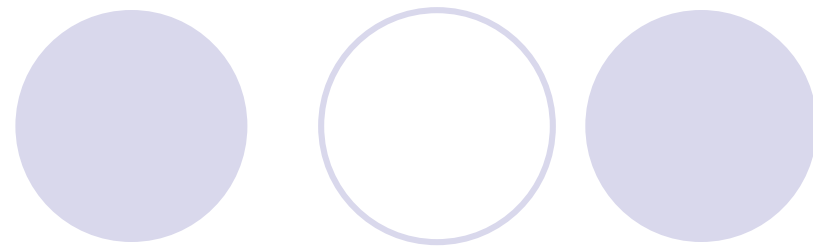
- 欠拟合 ( underfitting ) 和过拟合 ( Overfitting )
- 特征值丢失



# 总结：你应当知道的

- 适定的函数近似问题
  - 示例空间,  $X$
  - 标注的训练样本  $\{ \langle x_i, y_i \rangle \}$
  - 假设空间,  $H = \{ f: X \rightarrow Y \}$
- 学习是 $H$ 空间上的搜索/优化问题
  - 各种各样目标函数
    - 最小化训练误差 ( 0-1 损失 )
    - 在所有的满足最小误差的假设中, 选择最小的 ( ? )
- 决策树学习
  - 贪心的Top-down决策树
  - 过拟合以及剪枝
  - 扩展

# 思考问题（1）



- ID3和C4.5在树空间上执行启发式算法。为什么不穷尽搜索？



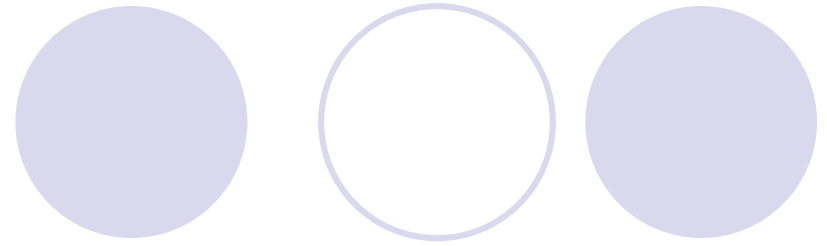
## 思考问题（2）

- 考虑目标函数  $f: \langle x_1, x_2 \rangle \rightarrow y$ ，这里  $x_1$  和  $x_2$  是实数，如果在树中每个属性只被允许使用一次，那么决策树可以表达的决策平面是什么样子的？

## 思考问题（3）

- 为什么用信息增益来选择属性？有没有其他方案

# 假设空间大小



- $n$ 布尔特征有多少个决策树

> 布尔函数函数个数

= 具有 $2^n$ 行的真值表个数 =  $2^{2^n}$

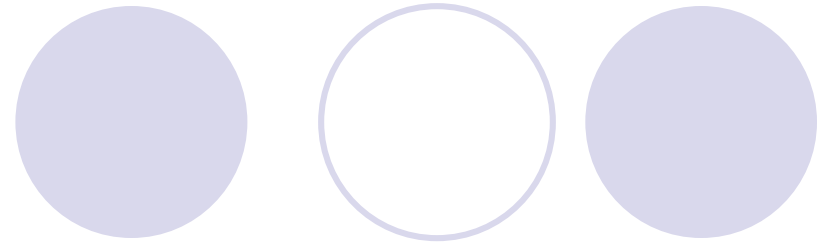
有6个布尔属性，则有

18,446,744,073,709,551,616棵树

# Notes on Overfitting

- **Overfitting results in decision trees that are more complex than necessary**
- **Training error no longer provides a good estimate of how well the tree will perform on previously unseen records**
- **Which Tree Should We Output?**
  - **Occam's razor: prefer the simplest hypothesis that fits the data**

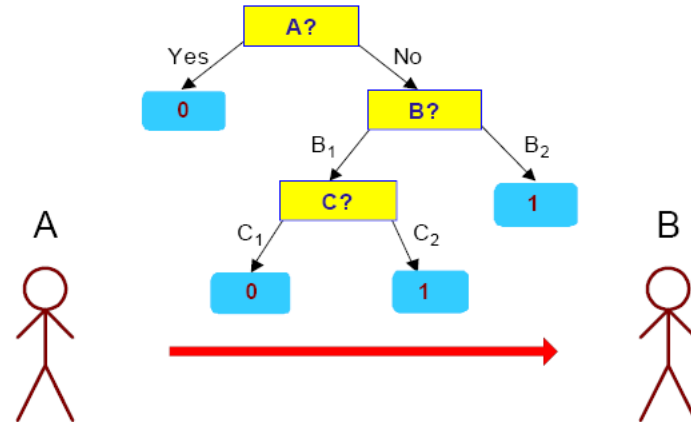
# Occam's Razor



- **Given two models of similar generalization errors, one should prefer the simpler model over the more complex model**
- **For complex models, there is a greater chance that it was fitted accidentally by errors in data**
- **Therefore, one should include model complexity when evaluating a model**

# Minimum Description Length (MDL)

X	y
X <sub>1</sub>	1
X <sub>2</sub>	0
X <sub>3</sub>	0
X <sub>4</sub>	1
...	...
X <sub>n</sub>	1



X	y
X <sub>1</sub>	?
X <sub>2</sub>	?
X <sub>3</sub>	?
X <sub>4</sub>	?
...	...
X <sub>n</sub>	?

- **$\text{Cost}(\text{Model}, \text{Data}) = \text{Cost}(\text{Data} | \text{Model}) + \text{Cost}(\text{Model})$** 
  - Cost is the number of bits needed for encoding.
  - Search for the least costly model.
- **$\text{Cost}(\text{Data} | \text{Model})$  encodes the misclassification errors.**
- **$\text{Cost}(\text{Model})$  uses node encoding (number of children) plus splitting condition encoding.**

# How to Address Overfitting

- **Pre-Pruning (Early Stopping Rule)**

- Stop the algorithm before it becomes a fully-grown tree

- Typical stopping conditions for a node:

- Stop if all instances belong to the same class
    - Stop if all the attribute values are the same



- More restrictive conditions:

- Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features (e.g., using  $\chi^2$  test)
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).

# How to Address Overfitting...

## ● **Post-pruning**

- Grow decision tree to its entirety
- Trim the nodes of the decision tree in a bottom-up fashion
- If generalization error improves after trimming, replace sub-tree by a leaf node.
- Class label of leaf node is determined from majority class of instances in the sub-tree
- Can use MDL for post-pruning



# Handling Missing Attribute Values

- **Missing values affect decision tree construction in three different ways:**
  - **Affects how impurity measures are computed**
  - **Affects how to distribute instance with missing value to child nodes**
  - **Affects how a test instance with missing value is classified**

# Computing Impurity Measure

Tid	Refund	Marital Status	Taxable Income	Class
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	?	Single	90K	Yes

Missing  
value

**Before Splitting:**

Entropy(Parent)

$$= -0.3 \log(0.3) - (0.7) \log(0.7) = 0.8813$$

	Class = Yes	Class = No
Refund=Yes	0	3
Refund=No	2	4
Refund=?	1	0

**Split on Refund:**

Entropy(Refund=Yes) = 0

Entropy(Refund=No)

$$= -(2/6) \log(2/6) - (4/6) \log(4/6) = 0.9183$$

Entropy(Children)

$$= 0.3 (0) + 0.6 (0.9183) = 0.551$$

$$\text{Gain} = 0.9 \times (0.8813 - 0.551) = 0.3303$$

# Distribute Instances

Tid	Refund	Marital Status	Taxable Income	Class
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

Refund									
Yes	No								
<table> <tr> <td>Class=Yes</td><td>0</td></tr> <tr> <td>Class=No</td><td>3</td></tr> </table>	Class=Yes	0	Class=No	3	<table> <tr> <td>Cheat=Yes</td><td>2</td></tr> <tr> <td>Cheat=No</td><td>4</td></tr> </table>	Cheat=Yes	2	Cheat=No	4
Class=Yes	0								
Class=No	3								
Cheat=Yes	2								
Cheat=No	4								

Tid	Refund	Marital Status	Taxable Income	Class
10	?	Single	90K	Yes

Refund									
Yes	No								
<table> <tr> <td>Class=Yes</td><td>0 + 3/9</td></tr> <tr> <td>Class=No</td><td>3</td></tr> </table>	Class=Yes	0 + 3/9	Class=No	3	<table> <tr> <td>Class=Yes</td><td>2 + 6/9</td></tr> <tr> <td>Class=No</td><td>4</td></tr> </table>	Class=Yes	2 + 6/9	Class=No	4
Class=Yes	0 + 3/9								
Class=No	3								
Class=Yes	2 + 6/9								
Class=No	4								

Probability that Refund=Yes is 3/9

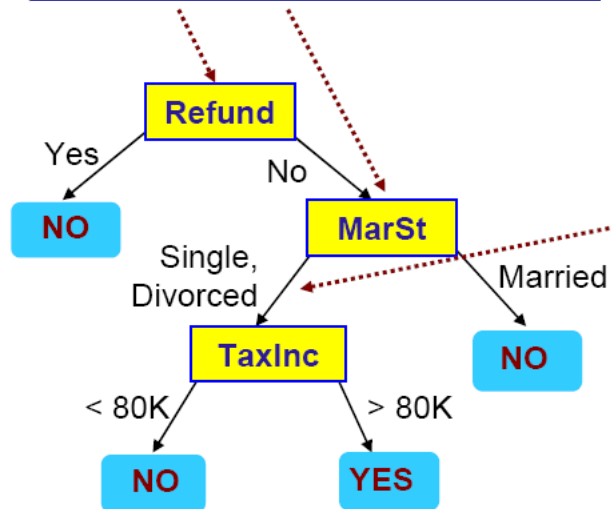
Probability that Refund=No is 6/9

Assign record to the left child with weight = 3/9 and to the right child with weight = 6/9

# Classify Instances

New record:

Tid	Refund	Marital Status	Taxable Income	Class
11	No	?	85K	?



	Married	Single	Divorced	Total
Class=No	3	1	0	4
Class=Yes	6/9	1	1	2.67
Total	3.67	2	1	6.67

Probability that Marital Status = Married is  $3.67/6.67$

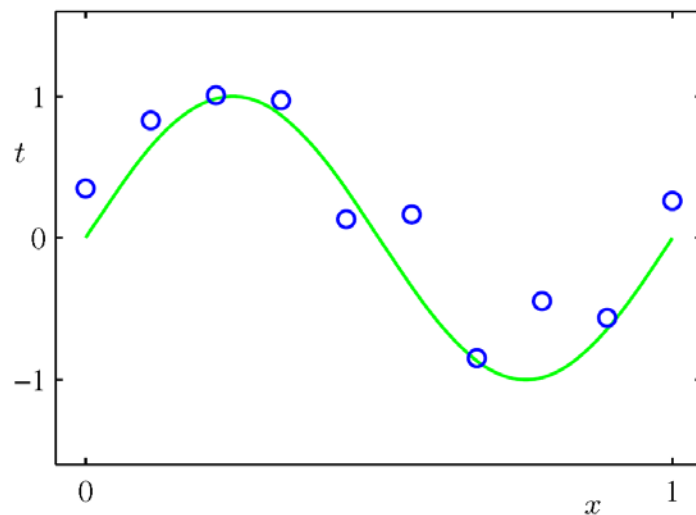
Probability that Marital Status = {Single, Divorced} is  $3/6.67$

# 例子：曲线拟合（回归问题）

- 问题：观测一个实数值 $x$ ，并用此值来预测目标值 $t$ 
  - 人工合成数据(便于研究)
  - $\sin(2\pi x)$
- 假设给定 $N$ 个观测数据的 *训练集合*
  - $X \equiv (x_1, \dots, x_n)^T$
  - $T \equiv (t_1, \dots, t_n)^T$

# 利用 $\sin(2\pi x)$ 产生样本

- $N=10$  ,  $x$ 均匀分布在 $[0,1]$
- 对每一个目标值 $t$ 加一个0均值的高斯噪声



# 问题的目标

- 利用训练集合
- 对每一个新 $\hat{x}$ ，预测目标值 $\hat{t}$
- 实质是学习正弦函数（即真实函数 $f$ ）
  - 这是非常困难的
  - 采用什么样的假设空间 $H$ ？
  - 多项式函数

$$y(x, w) = w_0 + w_1x + \cdots + w_mx^m = \sum_{i=0}^m w_i x^i$$

- $y(x, w)$ 是 $x$ 的多项式函数， $w$ 的线性函数

# 参数的确定（优化）

- 如何确定参数 $w$ （假设给定 $m$ ）

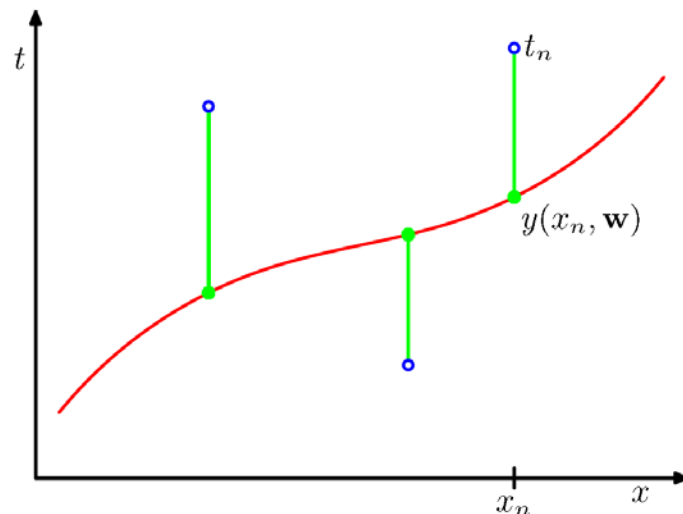
- 建立误差函数，测量每个样本点目标值 $t$ 与预测函数 $y$ 之间的误差

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$$

- 显然是最小二乘问题

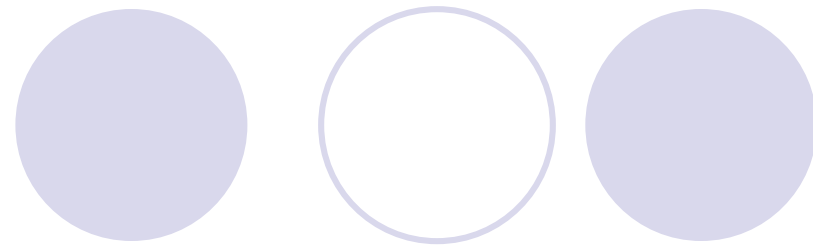
- $E$ 是 $w$ 的二次函数
- 求导，设倒数=0
- 存在唯一解 $w^*$

思考：为什么 $E$ 中有 $1/2$





# 多项式函数的阶

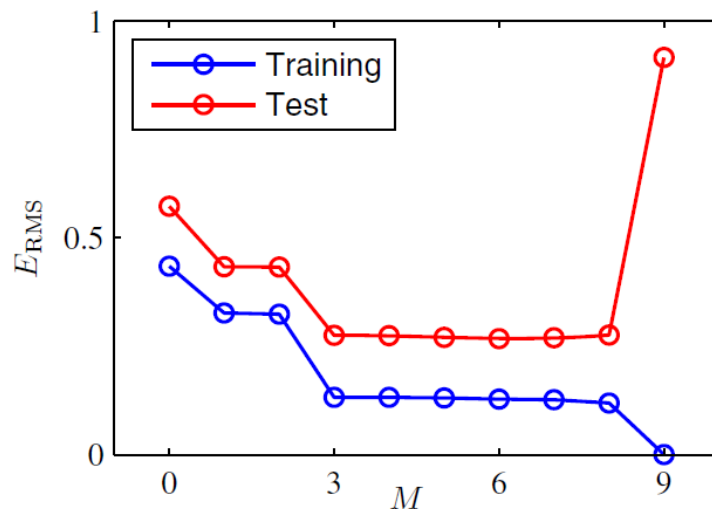
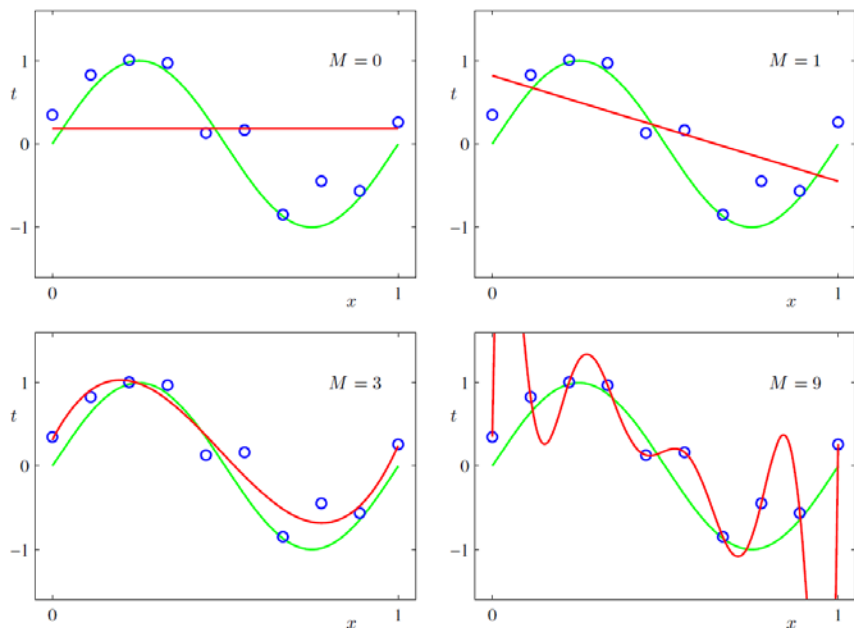


- 多项式函数阶的作用
  - 控制模型的表达能力
  - 模型的复杂（灵活）程度
- 多项式函数阶的确定
  - 模型比较
  - 模型选择

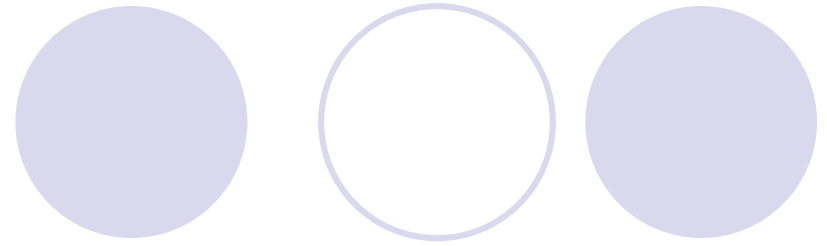
# 模型复杂度与过学习

- 阶数M取不同值时，最佳拟合曲线情况
- 拟合优度评价
  - 利用优化目标函数值
  - 方均根值（有时更方便）

$$E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$$

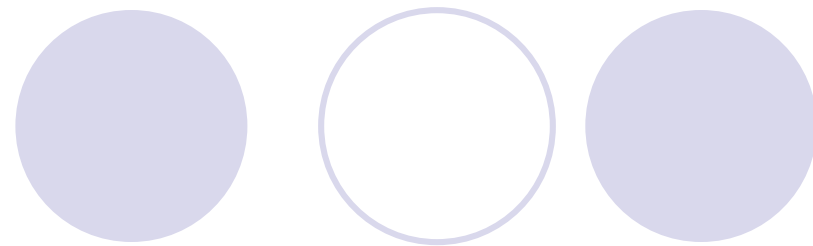


# 实验观察与分析



- 阶数 $M$ 较小时
  - 模型表达能力有限（不灵活）
  - 方均根误差大
- 阶数 $M=3-8$ 时
  - 拟合较好
  - 方均根误差较小
- 阶数 $M=9$ 时
  - 完全拟合
  - 误差为0
  - 但曲线已不具有正弦形状，为什么？
  - 能力最强

# 实验观察与分析



- N=10个样本

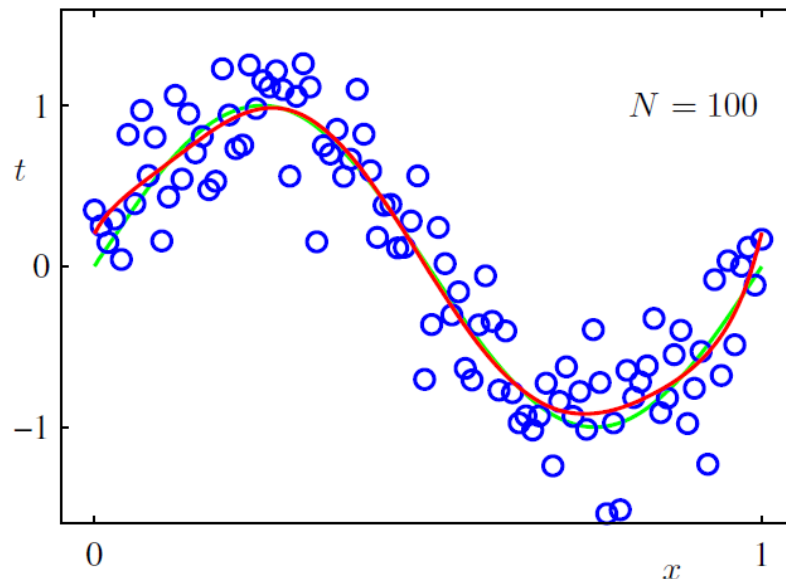
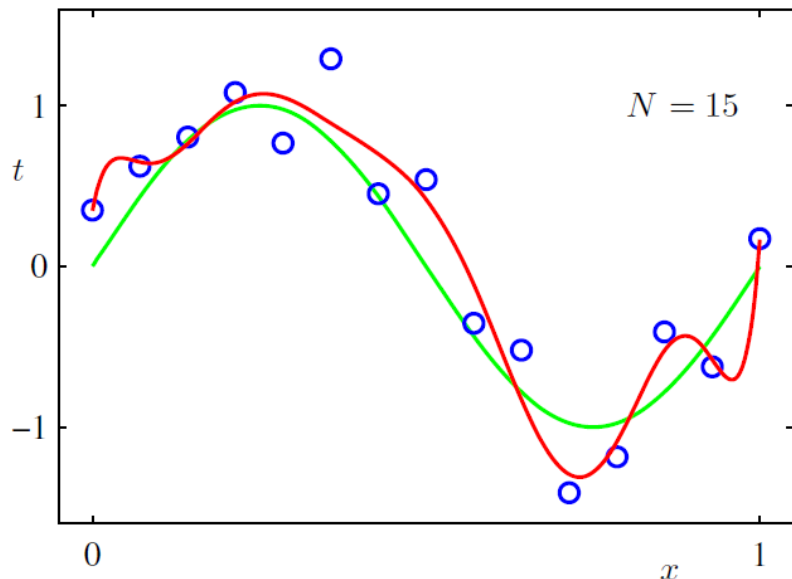
- M=9, 正好可以经过这N个训练样本

- 出现过学习

- 不同阶数时最佳拟合曲线参数 $w^*$ 的值

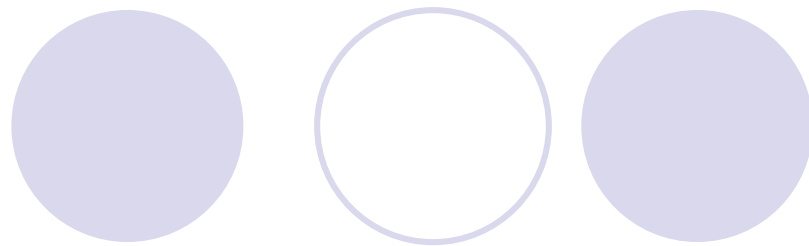
	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^*$	0.19	0.82	0.31	0.35
$w_1^*$		-1.27	7.99	232.37
$w_2^*$			-25.43	-5321.83
$w_3^*$			17.37	48568.31
$w_4^*$				-231639.30
$w_5^*$				640042.26
$w_6^*$				-1061800.52
$w_7^*$				1042400.18
$w_8^*$				-557682.99
$w_9^*$				125201.43

# 训练样本数量的作用



- $M=9$
- 增大样本数量可减少过学习程度

# 模型的复杂程度

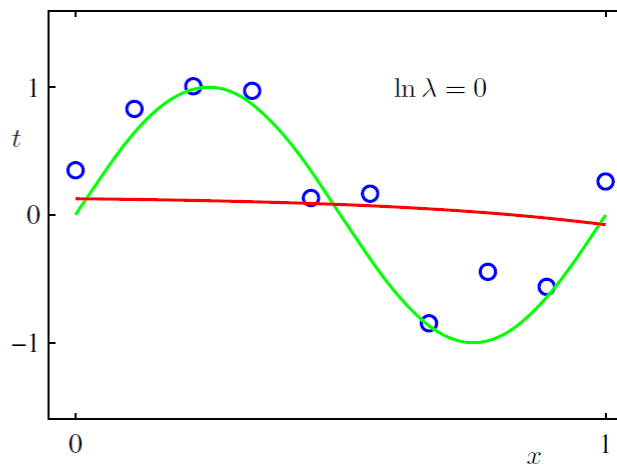
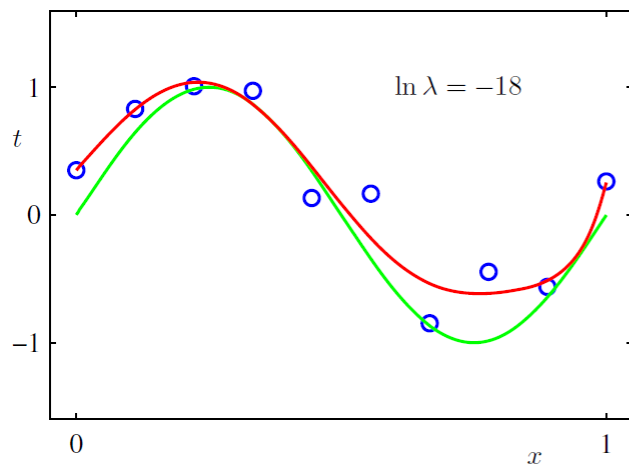


- 模型复杂度与什么有关？
  - 模型参数个数？
  - 模型参数 $w^*$ 的绝对值（范数）？
- 回想模型最佳参数 $w^*$ 的特点
  - 参数多时、 $w^*$ 往往具有大的绝对值
  - 可以在优化目标函数 $E(w)$ 中加入对 $w$ 的惩罚

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \underbrace{\frac{\lambda}{2} \|\mathbf{w}\|^2}$$

惩罚项或正则项

# 加入正则项后的实验



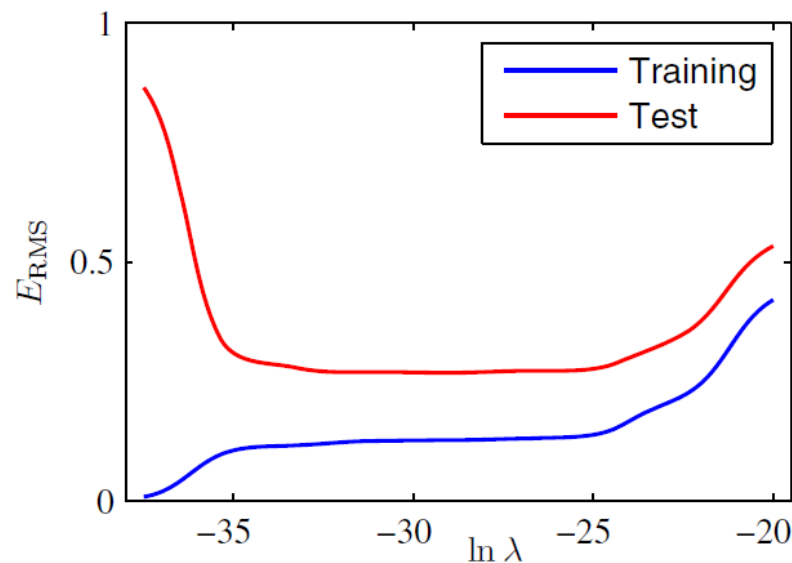
## ● 惩罚项比重影响

- 比重大时降低模型复杂度
- 比重适当时模型复杂度与问题匹配
- 比重小时、退化成原模型

# 加入正则项后的实验

## ● M=9时

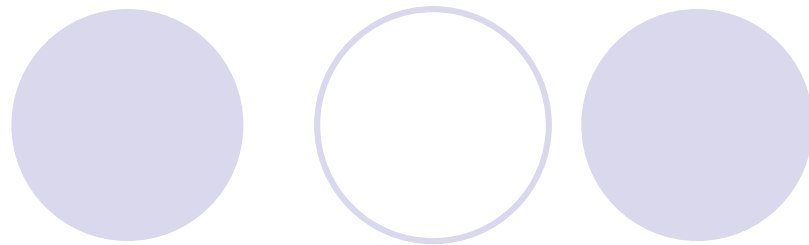
	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^*$	0.35	0.35	0.13
$w_1^*$	232.37	4.74	-0.05
$w_2^*$	-5321.83	-0.77	-0.06
$w_3^*$	48568.31	-31.97	-0.05
$w_4^*$	-231639.30	-3.89	-0.03
$w_5^*$	640042.26	55.28	-0.02
$w_6^*$	-1061800.52	41.32	-0.01
$w_7^*$	1042400.18	-45.95	-0.00
$w_8^*$	-557682.99	-91.53	0.00
$w_9^*$	125201.43	72.68	0.01



## ● 如何设置权重参数 $\lambda$ ？



# 权重参数 $\lambda$ 确定



- 实验方法

- 分为训练集合 ( training set )
- 验证集合 ( validation set )
- 测试集合 ( Test Set )