

---

# COMP9318: Data Warehousing and Data Mining

Assignment Project Exam Help

— L7: <https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

---

- Problem definition and preliminaries

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# ML Map

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Classification vs. Prediction

---

- Classification:
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set
  - classifying a new data <https://eduassistpro.github.io/>
- Prediction (aka)
  - models continuous-valued .e., predicts unknown or missing values
- Typical Applications
  - credit approval
  - target marketing
  - medical diagnosis
  - treatment effectiveness analysis

# Classification and Regression

---

- Given a new **object**  $\mathbf{o}$ , map it to a **feature vector**  $\mathbf{x} = (x_1, x_2, \dots, x_d)^\top$
- Predict the  $\mathbf{o}$  Assignment Project Exam Help  $y \in \mathcal{Y}$ 
  - Binary class <https://eduassistpro.github.io/> Sometimes,  $\{0, 1\}$   $\mathcal{Y} = \{-1, +1\}$   
Add WeChat edu\_assist\_pro
  - Multi-class classification  $\mathcal{Y} = \{1, 2, \dots, C\}$
- Learn a classification **function**:  $f(\mathbf{x}) : \mathbb{R}^d \mapsto \mathcal{Y}$
- Regression:  $f(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$

# Examples of Classification Problem

- Text categorization:

**Doc:** Months of campaigning  
and weeks of round-the-clock  
efforts in Iowa all came down to  
~~Assignment Project Exam Help~~  
a final push Su

**Topic:** {  
Politics  
Sport}

<https://eduassistpro.github.io/>

- Input object: sequence of words
  - Input features  $\mathbf{x} = \mathbf{w}$
  - Class label:  $y$
- Add WeChat edu\_assist\_pro
- freq(democrats)=2, freq(basketball)= 0, ...
  - $\mathbf{x} = [1, 2, 0, \dots]^T$
- 'Politics':  $y = +1$
- 'Sport':  $y = -1$

# Examples of Classification Problem

- Image Classification:



Assignment Project Exam Help

Which images are birds,  
which are not?

- Input object: <https://eduassistpro.github.io/>
- Input features  $\mathbf{x} = \text{Colo}$   
 $\text{Add WeChat } \text{edu\_assist\_pro}$ 
  - pixel\_count(red) = 1004, (blue) = 23000
  - $\mathbf{x} = [1004, 23000, \dots]^T$
- Class label  $y$ 
  - 'bird image':  $y = +1$
  - 'non-bird image':  $y = -1$

# How to find f()?

---

1. Input:
    - In supervised learning, we are given a set of training examples: Assignment: Project Exam Help
- <https://eduassistpro.github.io/>
- $$\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$$
- Identical independent distribution (i.i.d) assumption
    - A critical assumption for machine learning theory
    - e.g.,

$$\log P(\mathbf{x} \mid \theta) = \sum_i \log P(x_i \mid \theta)$$

# How to find $f()$ ?

---

2. Representation of  $f()$ 
  - Typically only consider a particular function family  $F$
  - Assignment Project Exam Help
    - consider  $f(\mathbf{x}; \theta) \in F$   
<https://eduassistpro.github.io/>

Examples: Add WeChat [edu\\_assist\\_pro](#)

- $F$ : linear functions (for regression)  $\rightarrow f = \mathbf{w}^T \mathbf{x}$
- $F$ : linear functions (for classification)  $\rightarrow f = \sigma(\mathbf{w}^T \mathbf{x})$
- What about more general function families?

# How to find $f()$ ?

## 3. Criterion for the best $f()$

- Non-Bayesian approaches:

ERM      • Loss function:  $\mathcal{L}(\{l(\hat{y}_i, y_i)\}) = \mathcal{L}(\{l(f(\mathbf{x}_i; \theta), y_i)\})$

SRM      • Regulariza

<https://eduassistpro.github.io/>

Examples:      Add WeChat edu\_assist\_pro

- Regression with L2 loss and L1 regularization

$$J(\boldsymbol{\theta}) = \sum_{i=1}^n (\hat{y}_i - y_i)^2 + \lambda \|\boldsymbol{\theta}\|_1$$

- Classification with cross entropy loss and L2 regularization

$$J(\boldsymbol{\theta}) = \sum_{i=1}^n \left( - \sum_{j=1}^k \mathbf{y}_{i,j} \log(\hat{\mathbf{y}}_{i,j}) \right) + \lambda \|\boldsymbol{\theta}\|_2^2$$

# Machine Learning Terminologies

- Supervised learning has input labelled data
  - $\# \text{instances} \times \# \text{attributes}$  matrix/table
  - $\# \text{attributes} = \# \text{features} + 1$ 
    - 1 (usu. the last attribute) is for the class attribute
- Labelled data split i
  - Training data <https://eduassistpro.github.io/> → Build a model
    - Training error =  $1.0 - \frac{\# \text{incorrect}}{\# \text{training}}$
  - Validation/development data → Select/refine the model
  - Testing data
    - Testing/generalization error =  $1.0 - \frac{\# \text{incorrect}}{\# \text{testing instances}}$
- We mainly discuss binary classification here → Evaluate the model
  - i.e.,  $\# \text{labels} = 2$

- 
- Overview of the whole process (not including cross-validation)

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Classification—A Two-Step Process

---

- **Model construction:** describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the **class label attribute**
  - The set of tuples used for model construction is **training set**
  - The model is represented by rules, decision trees, or mathematical <https://eduassistpro.github.io/>
- **Model usage:** for n objects
  - Estimate accuracy of the model
    - The known label of test sample and the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set, otherwise over-fitting will occur
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

# Classification Process (1): Preprocessing & Feature Engineering

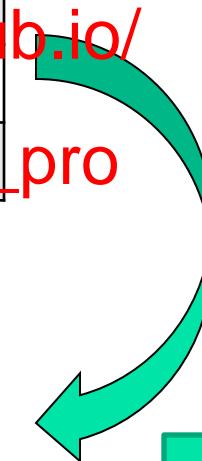
EID	Name	Title	EMP_Date	Track
101	Mike	Assistant Prof	2013-01-01	3-year contract
102	Mary	Assistant Prof	2009-04-15	Continuing
103	Bill	Scientia Professor	2014-02-03	Continuing
110	Jim	Associate Prof	2009-03-14	Continuing
121	Dave	Associate Prof		
234	Anne	Professor		
188	Sarah	Student Officer	2008-01-17	

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

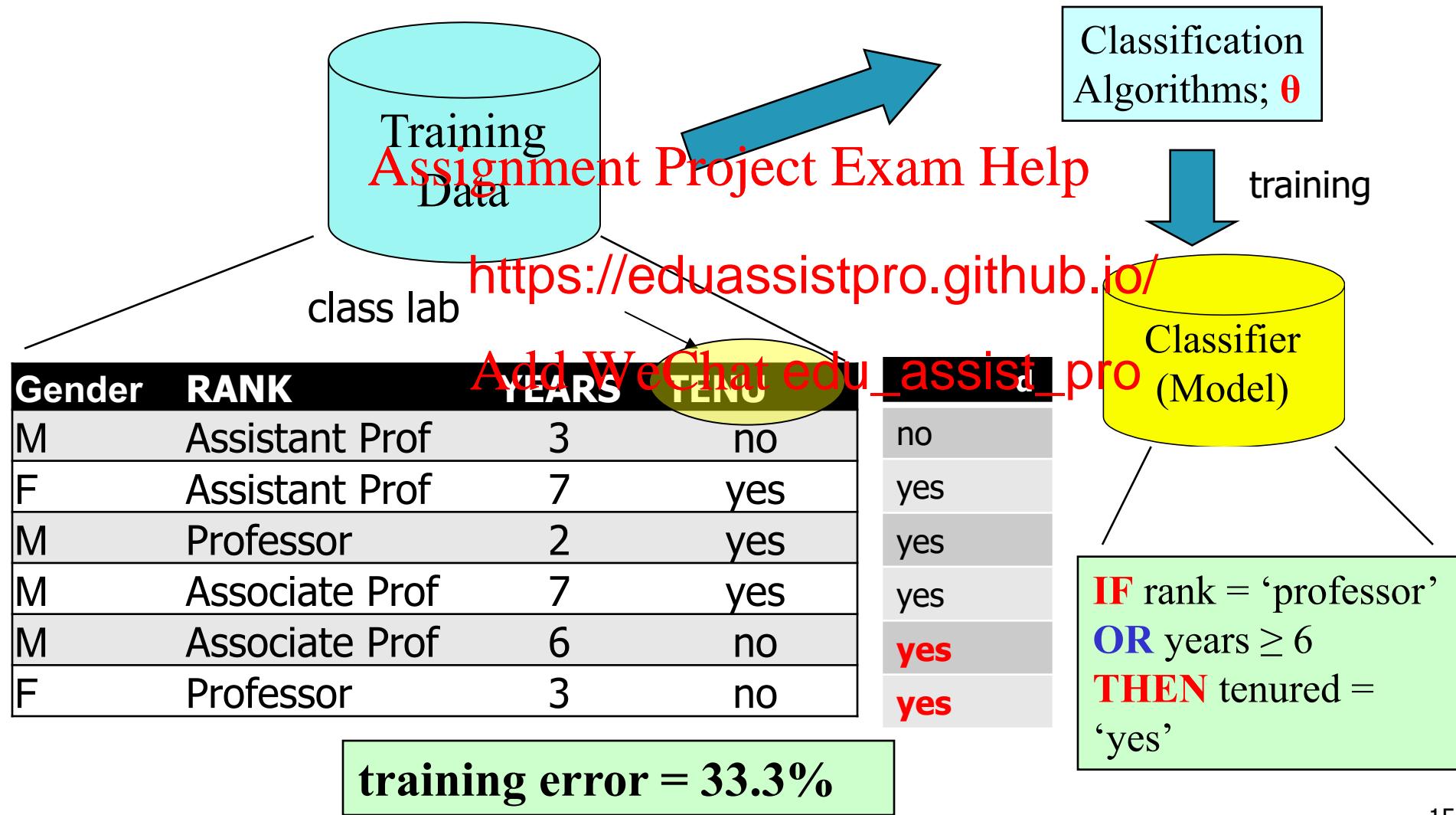
Raw  
Data



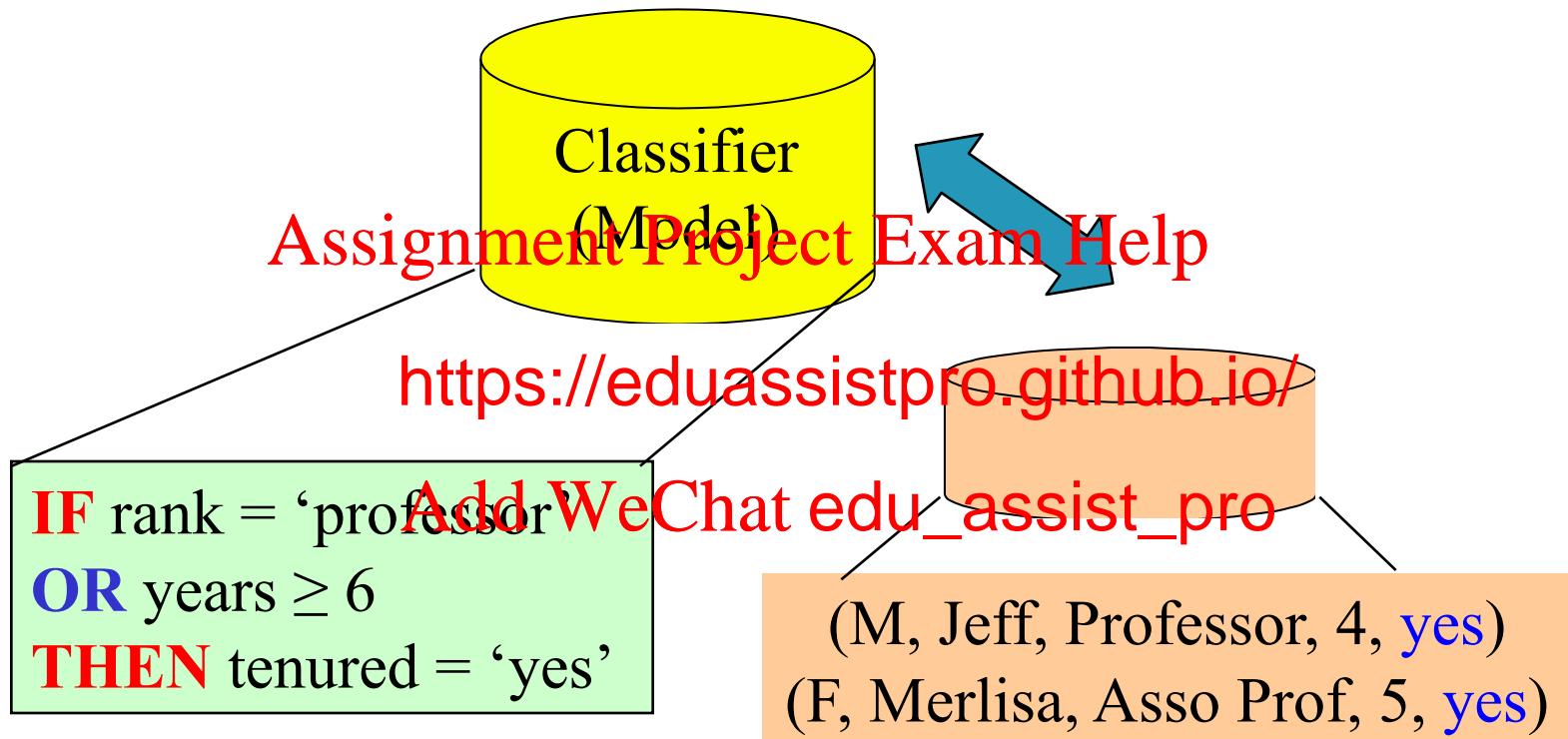
Gender	RANK	YEARS	TENURED
M	Assistant Prof	3	no
F	Assistant Prof	7	yes
M	Professor	2	yes
M	Associate Prof	7	yes
M	Associate Prof	6	no
F	Professor	3	no

Training  
Data

# Classification Process (2): Training

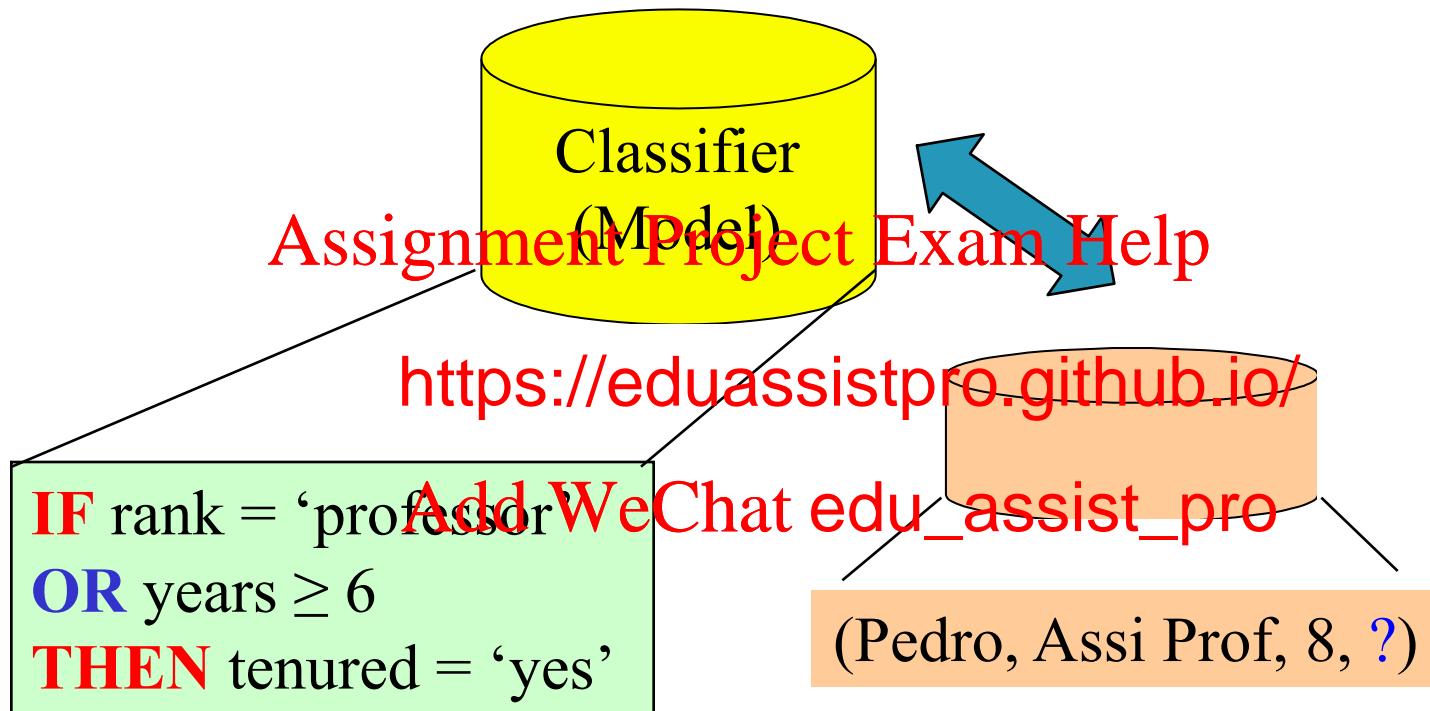


# Classification Process (3): Evaluate the Model on Testing Data



testing error = 50%

# Classification Process (4): Use the Model in Production



Yes

# How to judge a model?



## 10-fold CV



# Exercise: Problem definition and Feature Engineering

---

- How to formulate the following into ML problems?  
What kind of resources do you need? What are the features you think may be most relevant?  
**Assignment Project Exam Help**
  1. Predict the particular product in the next <https://eduassistpro.github.io/>  
**Add WeChat edu\_assist\_pro**
  2. Design an algorithm to produce better top-10 ranking results for queries

# Supervised vs. Unsupervised Learning

---

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class for each data point
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

- Decision Tree classifier
  - ID3
  - Other variants Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Training Dataset

This follows an example from Quinlan's ID3

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40				yes
>40				yes
>40			t	no
31...40	low	yes		yes
<=30	medium	no		no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

# Output: A Decision Tree for Computer Purchase

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Assignment Project Exam Help

student?

no

no

yes

yes

yes

age?

age	income	student	credit rating	buys computer
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
>40	medium	yes	fair	yes
>40	medium	no	excellent	no

credit rating?

excellent

no

fair

yes

age	income	student	credit rating	buys computer
31...40	high	no	fair	yes
31...40	low	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes

# Extracting Classification Rules from Trees

- Represent the knowledge in the form of **IF-THEN** rules
  - Rules are easier for humans to understand
- One rule is created for each path from the root to a leaf
  - Each attribute-value pair along a path forms a conjunction
  - The leaf node holds the class prediction
  - Example

**IF**  $age = "<=30"$  <https://eduassistpro.github.io/>

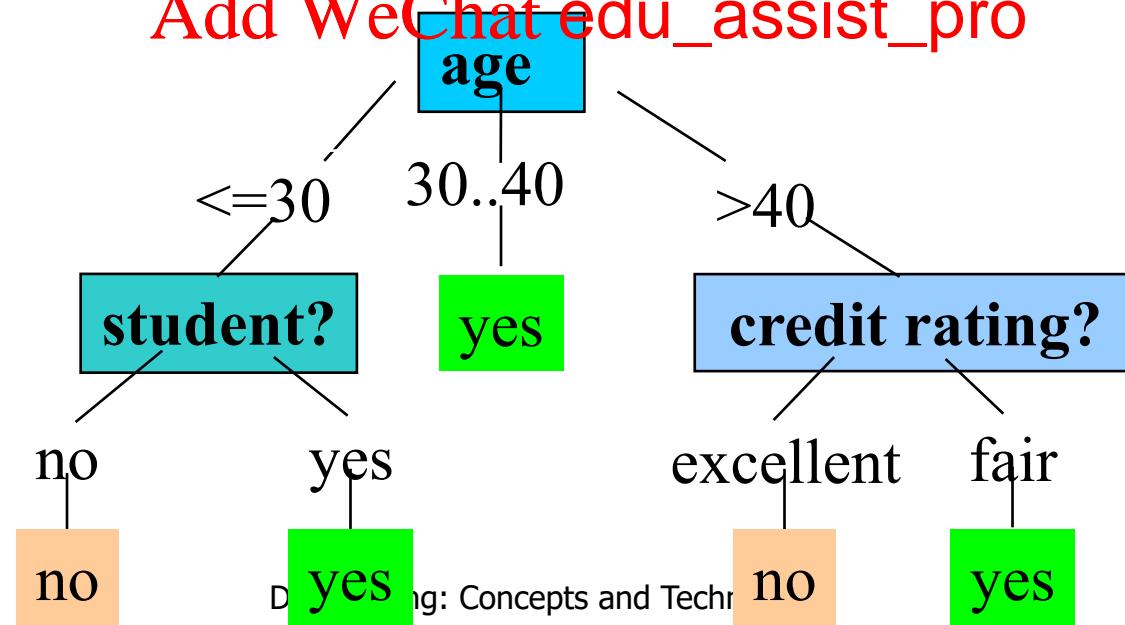
**IF**  $age = "<=30"$  **AND**  $student = "ye$

$ys\_computer = "no"$

$computer = "yes"$

Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

... ... ...



Exercise: Write down the pseudo-code of the induction algorithm

# Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Input: Attributes are categorical (if continuous-valued, they are **discretized** in advance)
  - Overview: Tree is constructed in a **top-down recursive divide-and-conquer manner**
    - At start, all samples are assigned to the root node
    - Samples are partitioned based on selected *test-attributes*
    - *Test-attributes* are selected according to a heuristic or statistical measure (e.g., **information gain**)
- Three conditions for stopping partitioning (i.e., boundary conditions)
  - There are no samples left, **OR**
  - All samples for a given node belong to the same class, **OR**
  - There are no remaining attributes for further partitioning (**majority voting** is employed for classifying the leaf)

# Decision Tree Induction Algorithm

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- S contains  $s_i$  tuples of class  $C_i$  for  $i = \{1, \dots, m\}$
- information measures info required to classify any arbitrary tuple

$$I(S_1, S_2, \dots) = -\sum_{i=1}^m \frac{s_i}{S} \log_2 \frac{s_i}{S}$$

- entropy of attribute A with val  $\{a_1, a_2, \dots, a_v\}$

$$E(A) = \sum_{j=1}^v \frac{s_{1j} + \dots + s_{mj}}{S} I(S_{1j}, \dots, S_{mj})$$

- information gained by branching on attribute A

$$Gain(A) = I(S_1, S_2, \dots, S_m) - E(A)$$

# Attribute Selection by Information Gain Computation

- Class P: buys\_computer = "yes"
- Class N: buys\_computer = "no"
- $I(p, n) = I(9, 5) = 0.940$
- Compute the entropy for *age*:

age	$p_i$	$n_i$	
$\leq 30$	2	3	
$30 \dots 40$	4	0	
$>40$	3	2	

$$E(\text{age}) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0)$$

$$+ \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$  means " $\text{age} \leq 30$ " has 5 out of 14 samples, with 2 yes's o's. Hence

$$I(p, n) = E(\text{age}) = 0.246$$

$$\text{Add WeChat } \text{edu\_assist\_pro}$$

$$me) = 0.029$$

$$Gain(\text{student}) = 0.151$$

$$Gain(\text{credit\_rating}) = 0.048$$

income	$p_i$	$n_i$	$I(p_i, n_i)$
high	2	2	1
medium	4	2	0.918

Q: what's the extreme/worst case?

age	income	student	credit_rating	buys_computer
$\leq 30$	high	no	fair	no
$\leq 30$	high	no	excellent	no
$31 \dots 40$	high	no	fair	yes
$>40$	medium	no	fair	yes
$>40$	low	yes	fair	yes
$>40$	low	yes	excellent	no
$31 \dots 40$	low	yes	excellent	yes
$\leq 30$	medium	no	fair	no
$\leq 30$	low	yes	fair	yes
$>40$	medium	yes	fair	yes
$\leq 30$	medium	yes	excellent	yes
$31 \dots 40$	medium	no	excellent	yes
$31 \dots 40$	high	yes	fair	yes
$>40$	medium	no	excellent	no

# Other Attribute Selection Measures and Splitting Choices

---

- Gini index (CART, IBM IntelligentMiner)
  - All attributes are assumed continuous-valued  
[Assignment](#) [Project](#) [Exam](#) [Help](#)
  - Assume the split values for each attribute <https://eduassistpro.github.io/>
  - May need other tools, such as Weka, to get the possible split values  
[Add WeChat](#) [edu\\_assist\\_pro](#)
  - Can be modified for categorical attributes
- Induces binary split => binary decision trees

# Gini Index (IBM IntelligentMiner)

- If a data set  $T$  contains examples from  $n$  classes, gini index,  $gini(T)$  is defined as

$$gini(T) = 1 - \sum_{j=1}^n p_j^2 = \sum p_j \cdot (1 - p_j)$$

where  $p_j$  is the proportion of class  $j$  in  $T$ .

- If a data set  $T$  is split into  $T_1$  and  $T_2$  with sizes  $N_1$  and  $N_2$  respectively, the gini index  $gini_{split}(T)$  is defined as

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- The attribute provides the smallest  $gini_{split}(T)$  is chosen to split the node (*need to enumerate all possible splitting points for each attribute*).

# Case I: Numerical Attributes

Age	Car	Class
20	...	Y
20	...	N
20	...	N
25	...	N
25	...	Y
30	...	Y
30	...	Y
30	...	Y
40	...	Y
40	...	Y

Split value

Cut=22.5	Y	N
<	1	2
$\geq$	6	1

$$Gini_{split}(S) = \frac{3}{10}Gini(1,2) + \frac{7}{10}Gini(6,1) = 0.30$$

Assignment Project Exam Help



<https://eduassistpro.github.io/>



27.5 Add WeChat edu\_assist\_pro

Cut=27.5	Y	N
		3
		3

$$Gini_{split}(S) = \frac{5}{10}Gini(2,3) + \frac{5}{10}Gini(5,0) = 0.24$$

...

...

$$\dots Gini(S) = 1 - \sum p_j^2$$

$$Gini_{split}(S) = \frac{n_1}{n} Gini(S_1) + \frac{n_2}{n} Gini(S_2)$$

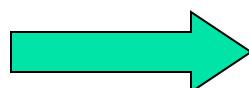
Exercise: compute gini indexes for other splits

# Case II: Categorical Attributes

count matrix

attrib list for Car

Age	Car	Class
20	M	Y
30	M	Y
25	T	N
30	S	Y
40	S	Y
20	T	N
30	M	Y
25	M	Y
40	M	Y
20	S	N



Assignment Project Exam Help

	Class=Y	Class=N
M	5	0
T	0	2
S	2	1

<https://eduassistpro.github.io/> for more splits !

	Y	N
{M, T}	5	2
{S}	2	1
{T}	0	2
{M}	5	0

$$\begin{aligned} \text{Gini}_{\text{split}}(S) &= \\ &7/10 * \text{Gini}(5,2) + \\ &3/10 * \text{Gini}(2,1) = \\ &0.42 \end{aligned}$$

$$\begin{aligned} \text{Gini}_{\text{split}}(S) &= \\ &8/10 * \text{Gini}(7,1) + \\ &2/10 * \text{Gini}(0,2) = \\ &0.18 \end{aligned}$$

$$\begin{aligned} \text{Gini}_{\text{split}}(S) &= \\ &5/10 * \text{Gini}(2,3) + \\ &5/10 * \text{Gini}(5,0) = \\ &0.24 \end{aligned}$$

# ID3

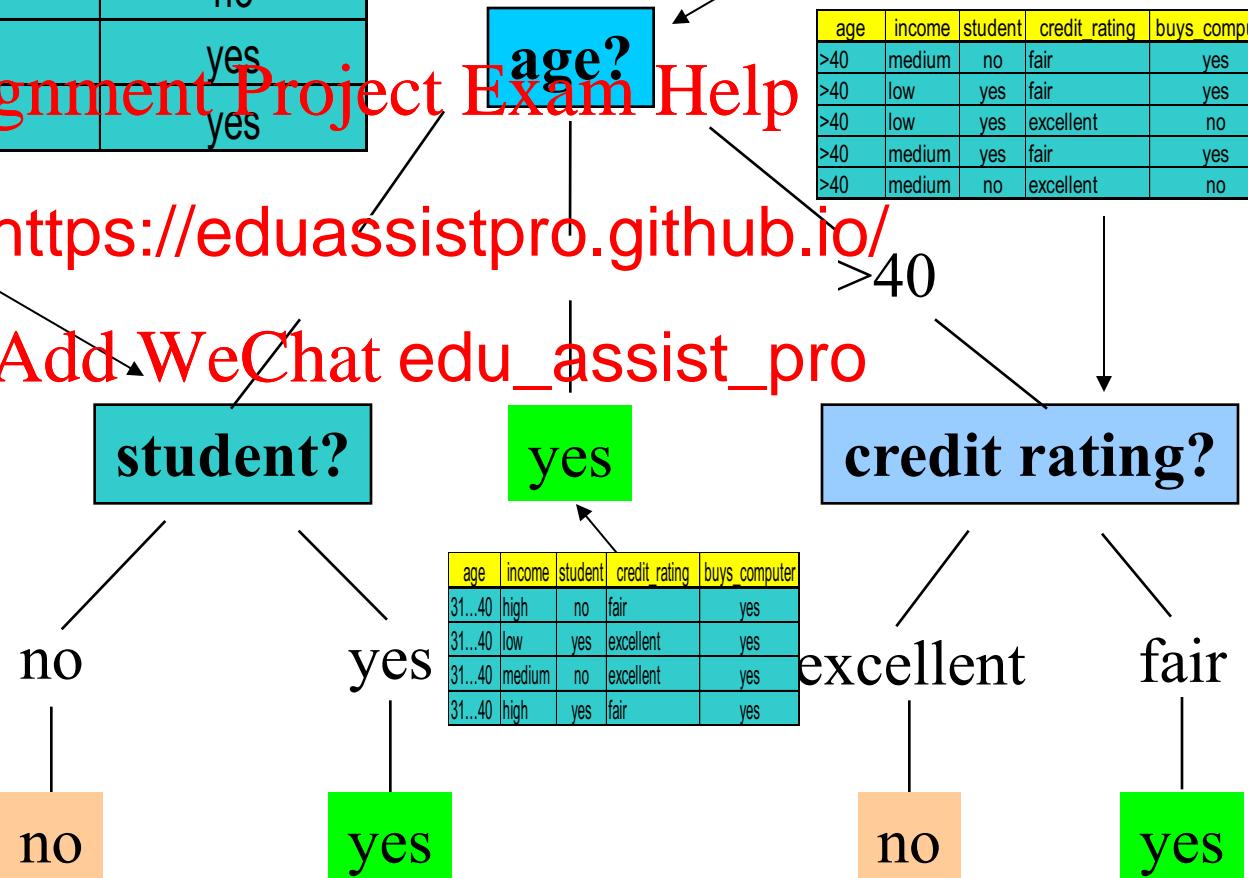
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

# Assignment Project Exam Help

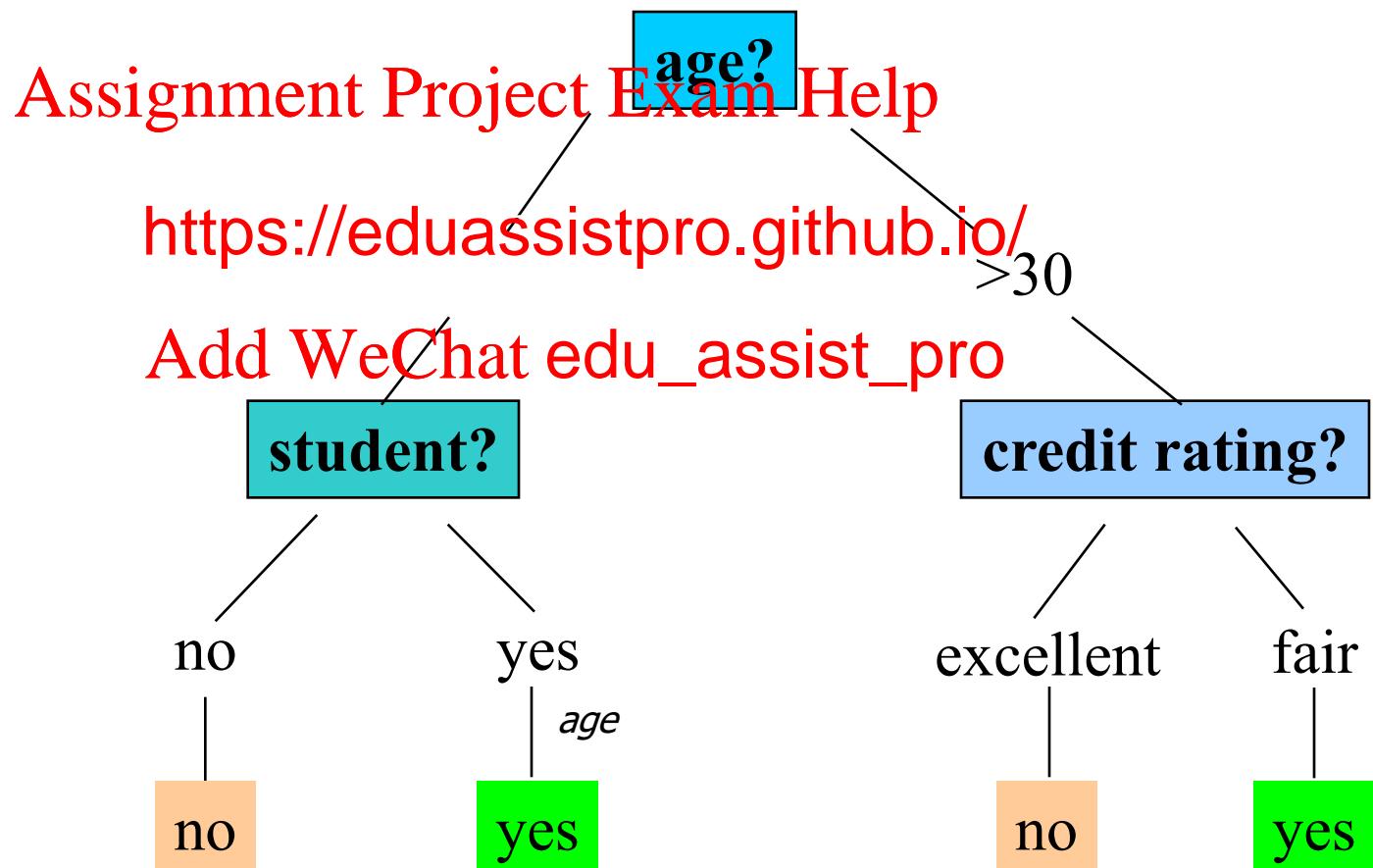
<https://eduassistpro.github.io/>

Add WeChat `edu_assist_pro`



# CART/SPRINT

Illustrative of  
the shape only



# Avoid Overfitting in Classification

---

- **Overfitting:** An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy
- Two approaches <https://eduassistpro.github.io/>
  - Prepruning: Halt tree construction of a node if this would result in falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: Remove branches from a “fully grown” tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the “best pruned tree”



- Lack of representative samples
- Existence of noise

# Overfitting Example /1

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Is Mammal?
Salamander	Cold-blooded	No	Yes	Yes	No
Guppy	Cold-blooded	Yes	No	No	No
Eagle	Warm-blooded	No	No	No	No
Poorwill	Warm-blooded			Yes	No
Platypus	Warm-blooded			Yes	Yes

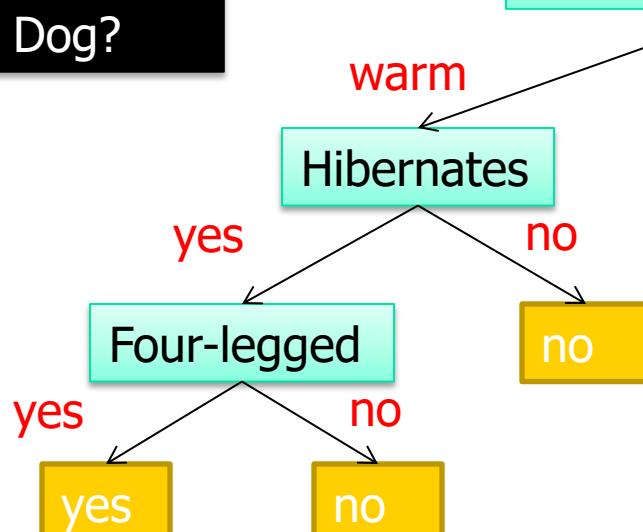
Assignment Project Exam Help

<https://eduassistpro.github.io/>



Human ?  
Dog?

Add WeChat edu\_assist\_pro



Training error = 0%, but  
does **not** generalize!





- Lack of representative samples
- Existence of noise

# Overfitting Example /2

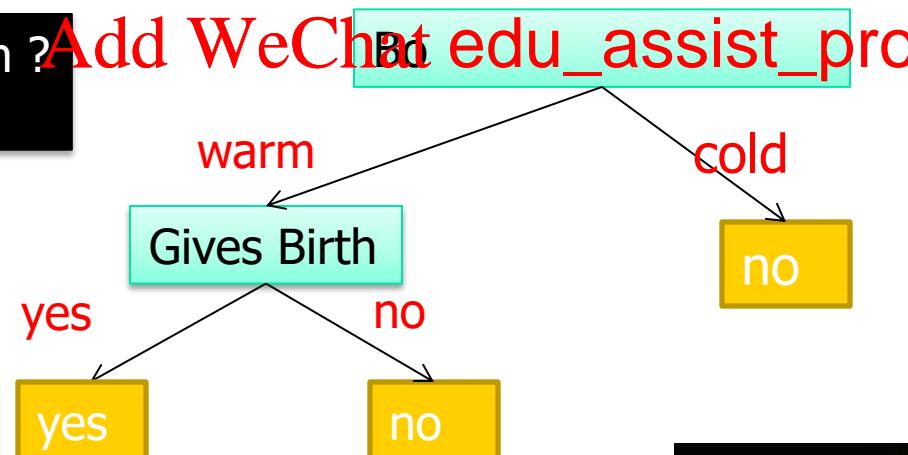
Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Is Mammal?
Salamander	Cold-blooded	No	Yes	Yes	No
Guppy	Cold-blooded	Yes	No	No	No
Eagle	Warm-blooded	No	No	No	No
Poorwill	Warm-blooded			Yes	No
Platypus	Warm-blooded			Yes	Yes

Assignment Project Exam Help

<https://eduassistpro.github.io/>



Human ?  
Dog?



Training error = 20%, but **generalizes!**

# Overfitting

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- Overfitting: model too complex → training error keep decreasing, but testing error increases
- Underfitting: model too simple → both training and testing has large errors.

# Overfitting

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Overfitting examples in Regression & Classification

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# DT Pruning Methods

---

- Use a separate validation set
- Estimation of generalization/test errors  
Assignment Project Exam Help
- Use all the data
  - but apply <https://eduassistpro.github.io/> -square) to estimate whether encoding or pruning a node may improve the entire distribution
- Use minimum description length (MDL) principle
  - halting growth of the tree when the encoding is minimized

# Pessimistic Post-pruning

Better estimate of generalization error in C4.5:  
*Use the upper 75% confidence bound from the training error of a node, assuming a binomial distribution*

- Observed on the training data
  - $e(t)$ : #errors on a leaf node  $t$  of the tree  $T$
  - $e(T) = \sum_{t \in T} e(t)$
- What's the generalization error (i.e. Errors on testing data) on  $T$ ?
  - Use pessimistic
  - $e'(t) = e(t) + 0.5N$ , where  $N$  is the number of leaf nodes in  $T$
  - $E'(t) = e(T) + 0.5N$ , where  $N$  is the number of leaf nodes in  $T$
- What's the generalization errors on  $r$ 
  - $E'(\text{root}(T)) = e(T) + 0.5$
- Post-pruning from bottom-up
  - If generalization error reduces after pruning, replace sub-tree by a leaf node
  - Use majority voting to decide the class label

# Example

Class = Yes	20
Class = No	10

$$\text{Error} = 10/30$$

- Training error before splitting on A =  $10/30$
  - Pessimistic error =  $(10+0.5)/30$
- Assignment Project Exam Help

Training Error after splitting on A

$$\hat{\mathcal{E}}_h$$

$$\hat{\mathcal{E}}_h$$

$$A?$$

$$4*0$$

$$/30$$

Add WeChat [edu\\_assist\\_pro](https://eduassistpro.github.io/)

Prune the subtree at A

A1

A2

A3

A4

Class = Yes	8
Class = No	4

Class = Yes	3
Class = No	4

Class = Yes	4
Class = No	1

Class = Yes	5
Class = No	1

# Classification in Large Databases

---

- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of classes  
<https://eduassistpro.github.io/>
- Why decision trees?
  - relatively faster learning speed than other classification methods)
  - convertible to simple and easy to understand classification rules
  - can use SQL queries for accessing databases
  - comparable classification accuracy with other methods

# Enhancements to basic decision tree induction

---

- Allow for continuous-valued attributes
  - Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals
- Handle missing values
  - Assign the most common value
  - Assign probability to each of the possible values
- Attribute construction
  - Create new attributes based on existing ones that are sparsely represented
  - This reduces fragmentation, repetition, and replication

---

## ■ Bayesian Classifiers

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Bayesian Classification: Why?

---

- Probabilistic learning: Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems  
**Assignment Project Exam Help**
- Incremental: E  
n incrementally  
increase/decrease hypothesis is  
correct. Prior k  
ned with observed  
data.  
**Add WeChat edu\_assist\_pro**
- Probabilistic prediction: Predict multiple hypotheses,  
weighted by their probabilities
- Standard: Even when Bayesian methods are  
computationally intractable, they can provide a standard  
of optimal decision making against which other methods  
can be measured

# Bayesian Theorem: Basics

---

- Let  $X$  be a data sample whose class label is **unknown**
- Let  $h$  be a **hypothesis** that  $X$  belongs to class C
- For classification problems, determine  $P(h|X)$ : the probability that <https://eduassistpro.github.io/> on the observed data sample  $X$
- $P(h)$ : **prior** probability of hypothesis  $h$ . the initial probability before we observe any data, reflects the background knowledge)
- $P(X)$ : probability that sample data is observed
- $P(X|h)$  : probability of observing the sample  $X$ , given that the hypothesis holds

# Bayesian Theorem

---

- Given training data  $X$ , *posteriori probability of a hypothesis*  $h$ ,  $P(h|X)$  follows the Bayes theorem

Assignment Project Exam Help

$$P(h | X) = \frac{P(X|h)P(h)}{P(X)}$$

<https://eduassistpro.github.io/>

- Informally, this  
posterior = likelihood  $\times$  prior / evidence
- MAP (**maximum posteriori**) hypothesis

$$h_{\text{MAP}} = \arg \max_{h \in H} P(h | X) = \arg \max_{h \in H} P(X|h)P(h)$$

- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

# Training dataset

**Hypotheses:**

$C_1$ : buys\_computer =  
'yes'

$C_2$ : buys\_computer =  
'no'

Data sample

$X = (\text{age} \leq 30,$

**Income**=medium,

**Student**=yes,

**Credit\_rating**=

Fair)

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30...40	high	no	fair	yes
				yes
				yes
				yes
>40	low		ent	no
31...40	low		ent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

# Sparsity Problem

$$h_{\text{MAP}} = \arg \max_{h \in H} P(X|h)P(h)$$

- Maximum likelihood Estimate of  $P(h)$ 
  - Let  $p$  be the probability that the class is  $C_1$
  - Consider a training set with  $n$  examples  $x_1, x_2, \dots, x_n$  and their corresponding labels  $y_1, y_2, \dots, y_n$
  - $L(x) = p^y(1-p)^{1-y}$
  - $L(X) = \prod_{x \in X} L(x)$
  - $I(X) = \log(L(X)) = \sum_{x \in X} y \log(p) + (1-y) \log(1-p)$
  - To maximize  $I(X)$ , let  $dI(X)/dp = 0 \Rightarrow p = (\sum y)/n$
  - $P(C_1) = 9/14, P(C_2) = 5/14$
- ML estimate of  $P(X|h) = ?$ 
  - Requires  $O(2^d)$  training examples, where  $d$  is the #features.

# Naïve Bayes Classifier

- Use a model
  - Assumption: attributes are **conditionally independent**:

$$P(X | C_i) = \prod_{k=1}^n P(x_k | C_i)$$

Assignment Project Exam Help

- The product of the probabilities of each element  $x_1$  and  $x_2$ , given the current class  $C_i$  is the product of the probabilities of each element separately, given the same class  $C_i$ :  $P([x_1, x_2], C_i) = P(x_1 | C_i) \cdot P(x_2 | C_i)$
- No dependence relation between attributes given the class
- Greatly reduces the computation cost, only count the class distribution → Only need to estimate  $P(x_k | C_i)$

# Naïve Bayesian Classifier: Example

- Compute  $P(X|C_i)$  for each class

$X=(\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit\_rating} = \text{fair})$

$P(\text{age} = \text{"<30"}) | \text{Assignment Project Exam Help}$

$P(\text{age} = \text{"<30"}) | \text{buys\_}$

$P(\text{income} = \text{"medium"})$

$P(\text{income} = \text{"medium"})$

$P(\text{student} = \text{"yes"}) | \text{buys\_computer} = \text{"yes"} =$

$P(\text{student} = \text{"yes"}) | \text{buys\_computer} = \text{"no"} =$

$P(\text{credit\_rating} = \text{"fair"}) | \text{buys\_computer} = \text{"y}$

$P(\text{credit\_rating} = \text{"fair"}) | \text{buys\_computer} = \text{"no"} = 2/5 = 0.4$

<https://eduassistpro.github.io/>

$P(X | C_i) : P(X | \text{buys\_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.0.667 = 0.044$

$P(X | \text{buys\_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$

$P(X | C_i) * P(C_i) : P(X | \text{buys\_computer} = \text{"yes"}) * P(\text{buys\_computer} = \text{"yes"}) = 0.028$

$P(X | \text{buys\_computer} = \text{"no"}) * P(\text{buys\_computer} = \text{"no"}) = 0.007$

$X$  belongs to class "buys\_computer=yes"

likelihood

# The Need for Smoothing

---

- $\Pr[X_i = v_j \mid C_k]$  could still be 0, if not observed in the training data
  - makes  $\Pr[C_k \mid X] = 0$ , regardless of other likelihood  $v_k]$
- Add-1 Smooth
  - reserve a small amount for unseen probabilities
  - (conditional) probabilities of observed events have to be **adjusted** to make the total probability equals 1.0

# Add-1 Smoothing

---

- $\Pr[X_i = v_j \mid C_k] = \text{Count}(X_i = v_j, C_k) / \text{Count}(C_k)$
- $\Pr[X_i = v_j \mid C_k] = [\text{Count}(X_i = v_j, C_k) + 1] / [\text{Count}(C_k) + B]$   
**Assignment Project Exam Help**
- What's the right <https://eduassistpro.github.io/>
  - make  $\sum_{v_j} \Pr[X_i = v_j \mid C_k] = 1$
  - Explain the above constraint
    - $\Pr[X_i \mid C_k]$ : Given  $C_k$ , the (conditional) probability of  $X_i$  taking a specific value
    - $X_i$  must take one of the values, hence summing over all possible values for  $X_i$ , the probabilities should sum up to 1.0
  - $B = \text{dom}(X_i)$ , i.e., # of values  $X_i$  can take

# Smoothing Example

no instance of age  $\leq 30$  in the "No" class



Class:

C1:`buys_computer='yes'`  
C2:`buys_computer='no'`

Data sample

X = (**age**  $\leq 30$ ,  
**Income**=medium,  
**Student**=yes,  
**Credit\_rating**=  
Fair)

age	income	student	credit_rating	buys_computer
30...40	high	no	fair	no
30...40	high	no	excellent	no
3		r		yes
>		r		yes
>		r		yes
>40	low		lent	no
31...40	low		lent	yes
30...40	medium	no	fair	no
$\leq 30$	low	yes	fair	yes
>40	medium	yes	fair	yes
$\leq 30$	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

B=3

B=3

B=2

B=2

# Consider the “No” Class

- Compute  $P(X|C_i)$  for the “No” class

**X=(age<=30 , income =medium, student=yes, credit\_rating=fair)**

$$P(\text{age} = \text{"<30"} | \text{buys\_computer} = \text{"no"}) = (0 + 1) / (5 + 3) = 0.125$$

Assignment Project Exam Help

$$P(\text{income} = \text{"medium"} | b + 3) = 0.375$$

$$P(\text{student} = \text{"yes"} | \text{buys\_}) \quad \text{https://eduassistpro.github.io/} \quad 0.286$$

$$P(\text{credit\_rating} = \text{"fair"} | \text{buys\_computer} = \text{"no"}) = 0.429$$

$$\mathbf{P(X|Ci)} : P(X|\text{buys\_computer} = \text{"no"}) = 0.125 \times 0.375 \times 0.286 \times 0.429 = 0.00575$$

$$\mathbf{P(X|Ci)*P(Ci)} : P(X|\text{buys\_computer} = \text{"no"}) * P(\text{buys\_computer} = \text{"no"}) = 0.00205$$

Probabilities	Without Smoothing	With Smoothing
$\Pr[<=30   \text{No}]$	0 / 5	1 / 8
$\Pr[30..40   \text{No}]$	3 / 5	4 / 8
$\Pr[>=40   \text{No}]$	2 / 5	3 / 8

# How to Handle Numeric Values

---

- Need to model the distribution of  $\Pr[X_i | C_k]$
- Method 1:
  - Assume ~~Assignment Project Exam Help~~ Gaussian Naïve Bayes
  - $\Pr[X_i = v_j | C_k] \propto \frac{1}{\sqrt{2\pi} \sigma_i} e^{-\frac{(v_j - \mu_{ik})^2}{2\sigma_i^2}}$
- Method 2:
  - Use binning to discretize the feature values

# Text Classification

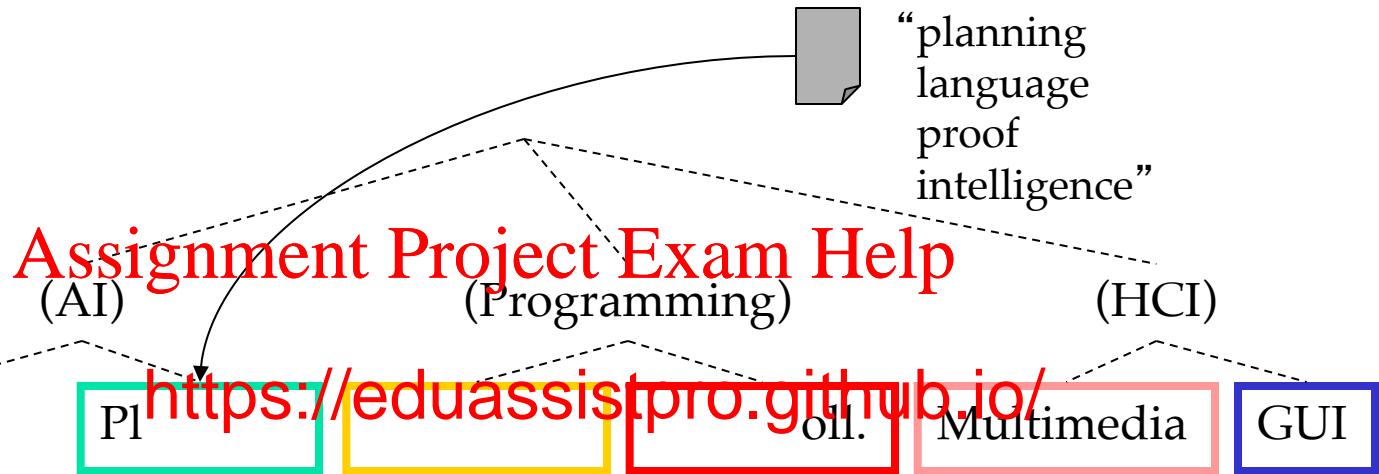
---

- NB has been widely used in **text classification** (aka., text categorization)
- Outline: Assignment Project Exam Help
  - Application <https://eduassistpro.github.io/>
  - Language [Add WeChat edu\\_assist\\_pro](#)
  - Two Classification Meth

Based on “Chap 13: Text classification & Naive Bayes”  
in Introduction to Information Retrieval  
• <http://nlp.stanford.edu/IR-book/>

# Document Classification

*Test  
Data:*



*Classes:*

*Training  
Data:*

learning	planning	program	... ...
<u>intelligence</u>	temporal	semantics	collection
algorithm	reasoning	<u>language</u>	memory
reinforcement	plan	<u>proof...</u>	optimization
network...	<u>language...</u>		region...

(Note: in real life there is often a hierarchy, not present in the above problem statement; and also, you get papers on ML approaches to Garb. Coll.)

# More Text Classification Examples:

## Many search engine functionalities use classification

---

Assign labels to each document or web-page:

- Labels are most often topics such as Yahoo-categories
  - e.g., "finance," "sports," "news>world>asia>business"
- Labels may be genres
  - e.g., "editorials" "
- Labels may be opinions
  - e.g., "like", "hate" "neutral"
- Labels may be domain-specific
  - e.g., "interesting-to-me": "not-interesting-to-me"
  - e.g., "contains adult language": "doesn't"
  - e.g., *language identification: English, French, Chinese, ...*
  - e.g., *search vertical: about Linux versus not*
  - e.g., "link spam": "not link spam"

# Challenge in Applying NB to Text

- Need to model  $P(\text{text} | \text{class})$ 
  - e.g.,  $P(\text{"a b c d"} | \text{class})$
- Extremely sparse → Need a model to help  
**Assignment Project Exam Help**
- 1<sup>st</sup> method:
  - Based on a s <https://eduassistpro.github.io/>
    - Turns out to be the **bag** of w using unigrams
    - → multinomial NB
- 2<sup>nd</sup> method:
  - View a text as a **set** of tokens → a Boolean vector in  $\{0, 1\}^{|V|}$ , where V is the vocabulary.
  - → Bernoulli NB

# Unigram and higher-order Language models in Information Retrieval

---

- $P("a\ b\ c\ d") =$ 
    - $P(a) * P(b | a) * P(c | a\ b) * P(d | a\ b\ c)$
  - Unigram model: 0-th order Markov Model
    - $P(d | a\ b\ c)$  <https://eduassistpro.github.io/>
  - Bigram Language Model
    - $P(d | a\ b\ c) = P(d | c)$
    - $P(a\ b\ c\ d) = P(a) * P(b | a) * P(c | b) * P(d | c)$
  - The same with class-conditional probabilities,  
i.e.,  $P("a\ b\ c\ d" | C)$
- Easy.  
Effective!

# Two Models /1

---

- Model 1: Multinomial = Class conditional unigram
  - One feature  $X_i$  for each word pos in document
    - feature's values are all words in dictionary
  - Value of  $X_i$  is the word in position  $i$
  - Naïve Bayes
    - Given the document tells us nothing about positions in the other positions
  - Second assumption:
    - Word appearance does not depend on position

$$P(X_i = w | c) = P(X_j = w | c)$$

for all positions  $i, j$ , word  $w$ , and class  $c$

- Just have one multinomial feature predicting all words

# Using **Multinomial** Naive Bayes Classifiers to Classify Text: Basic method

$$P(C_j | w_1 w_2 w_3 w_4) \propto P(C_j) P(w_1 w_2 w_3 w_4 | C_j)$$

an example  
text of 4  
tokens

$$\propto P(c_j) \prod_{i=1}^4 P(w_i | C_j)$$

Assignment Project Exam Help

unigram  
model

<https://eduassistpro.github.io/>

bag of word;  
multinomial

- Essentially, the classification is independent of the positions of the words

- Use same parameters for each position
- Result is **bag** of words model, i.e. text  $\equiv \{x_i : \theta_i\}_{i=1}^{|V|}$

e.g., "to be or not to be"

# Naïve Bayes: Learning

---

- From training corpus, extract  $V = \text{Vocabulary}$
- Calculate required  $P(c_j)$  and  $P(x_k | c_j)$  terms
  - For each  $c_j$  in  $C$  do
    - $docs_j \leftarrow$  all documents in  $C$  for which the target class is <https://eduassistpro.github.io/>
    - $P(c_j) \leftarrow \frac{|docs_j|}{\text{total # documents}}$
  - $Text_j \leftarrow$  single document containing all  $docs_j$
  - for each word  $x_k$  in  $\text{Vocabulary}$ 
    - $n_k \leftarrow$  number of occurrences of  $x_k$  in  $Text_j$
    - $P(x_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha | \text{Vocabulary} |}$

# Naïve Bayes: Classifying

---

- positions  $\leftarrow$  all word positions in current document which contain tokens found in *Vocabulary*  
**Assignment Project Exam Help**
- Return  $c_{NB}$ , whe  
<https://eduassistpro.github.io/>

Add WeChat **edu\_assist\_pro**

$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in positions} P(x_i | c_j)$$

# Naive Bayes: Time Complexity

---

- **Training Time:**  $O(|D|L_d + |C||V|)$

where  $L_d$  is the average length of a document in  $D$ .

Assignment Project Exam Help

- Assumes  $V$  and all  $D_i$ ,  $n_i$ , and  $n_{ij}$  pre-computed in  $O(|D|L_d)$  time during <https://eduassistpro.github.io/> ata.
- Generally just  $O(|D|L_d)$  since  $< |D|L_d$

Why?

- **Test Time:**  $O(|C| L_t)$

where  $L_t$  is the average length of a test document.

- Very efficient overall, linearly proportional to the time needed to just read in all the data.

Add WeChat edu\_assist\_pro

# Bernoulli Model

---

- $V = \{a, b, c, d, e\} = \{x_1, x_2, x_3, x_4, x_5\}$
- Feature functions  $f_i(\text{text}) = \text{if text contains } x_i$
- The feature functions extract a vector of  $\{0, 1\}^{|V|}$  from any tex <https://eduassistpro.github.io/>
- Apply NB directly [Assignment Project Exam Help](#) [Add WeChat edu\\_assist\\_pro](#)

"a b c d"
"d e b"



a	b	c	d	e	C
1	1	1	1	0	+
0	1	0	1	1	-

# Two Models /2

---

- Model 2: Multivariate Bernoulli
  - One feature  $X_w$  for each word in dictionary
  - $X_w = \text{true}$  if word  $w$  appears in document  $d$
  - Naive Bayes <https://eduassistpro.github.io/>
    - Given the appearance of one word in the document tell about chances that another word appears
- This is the model used in the binary independence model in classic probabilistic relevance feedback in hand-classified data (Maron in IR was a very early user of NB)

# Parameter estimation

---

- Multivariate Bernoulli model:

$\hat{P}(X_w = t | c_j) = \frac{\text{fraction of documents of topic } c_j \text{ in which word } w \text{ appears}}{\text{Assignment Project Exam Help}}$

<https://eduassistpro.github.io/>

- Multinomial model:

Add WeChat edu\_assist\_pro

$\hat{P}(X_i = w | c_j) = \frac{\text{fraction of times in which word } w \text{ appears across all documents of topic } c_j}{\text{ }}$

- Can create a mega-document for topic  $j$  by concatenating all documents in this topic
- Use frequency of  $w$  in mega-document

# Classification

---

- Multinomial vs Multivariate Bernoulli?  
  
**Assignment Project Exam Help**
- Multinomial  
ays more effective in <https://eduassistpro.github.io/>
  - See results figures lat  
**Add WeChat edu\_assist\_pro**
- See *IIR* sections 13.2 and 13.3 for worked examples with each model

# NB Model Comparison: WebKB

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Underflow Prevention: log space

---

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since  $\log(xy) = \log(x) + \log(y)$ , it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.  
<https://eduassistpro.github.io/>
- Class with highest probability is still the most probable

$$c_{NB} = \operatorname{argmax}_{c_j \in C} \log P(c_j) + \sum_{i \in positions} \log P(x_i | c_j)$$

- Note that model is now just max of sum of weights...

# Violation of NB Assumptions

---

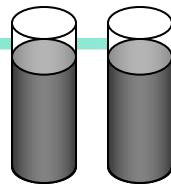
- **Conditional independence**
- “Positional independence”
- Examples [Assignment](#) [Project](#) [Exam](#) [Help](#)

<https://eduassistpro.github.io/>

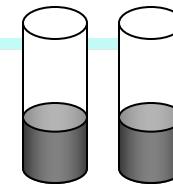
Add WeChat `edu_assist_pro`

# Observations

Raining



Sunny



$$P(+,+,\text{r}) = 3/8 \quad P(-,-,\text{r}) = 1/8$$

$$P(+,+,\text{s}) = 1/8 \quad P(-,-,\text{s}) = 3/8$$

~~Assignment Project Exam Help~~

- Two identical sensors at same location

<https://eduassistpro.github.io/>

- Equivalent training dataset

~~Add WeChat edu\_assist\_pro~~

- Note:  $P(s_1|C) = P(s_2|C)$  no matter

$$P(s_i = + | \text{r}) =$$

$$P(s_i = - | \text{r}) =$$

$$P(s_i = + | \text{s}) =$$

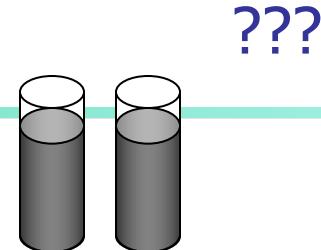
$$P(s_i = - | \text{s}) =$$

$$P(\text{r}) =$$

$$P(\text{s}) =$$

S1	S2	Class
+	+	r
+	+	r
-	-	r
+	+	s
-	-	s
-	-	s
-	-	s

# Prediction



Assignment Project Exam Help

- $P(r | ++) \propto$

<https://eduassistpro.github.io/> + ???

- $P(s | ++) \propto$

Add WeChat edu\_assist\_pro

$$P(s_i = + | r) = 3/4$$

$$P(s_i = - | r) = 1/4$$

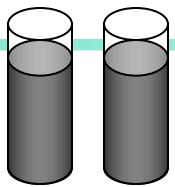
$$P(r) = 1/2$$

$$P(s) = 1/2$$

$$P(s_i = + | s) = 1/4$$

$$P(s_i = - | s) = 3/4$$

???



Problem: Posterior probability estimation not accurate

Reason: Correlation between features

Fix: Use logistic regression classifier

## Assignment Project Exam Help

- $P(r |++) \propto 9/3$

<https://eduassistpro.github.io/>

- $P(s |++) \propto 1/32$

Add WeChat edu\_assist\_pro

$$P(s_i = + | r) = 3/4$$

$$P(s_i = - | r) = 1/4$$

$$P(s_i = + | s) = 1/4$$

$$P(s_i = - | s) = 3/4$$

$$P(r) = 1/2$$

$$P(s) = 1/2$$

	S1	S2	Class
/10	-	+	r
10	+	+	r
	+	+	r
	-	-	r
	+	+	s
	-	-	s
	-	-	s
	-	-	s

# Naïve Bayes Posterior Probabilities

---

- Classification results of naïve Bayes (the class with maximum posterior probability) are usually fairly accurate.  
*Assignment Project Exam Help*
- However, due to the conditional independence of the features, the estimated posterior probabilities are often very small or very large.  
*https://eduassistpro.github.io/*  
*Add WeChat edu\_assist\_pro*
  - Output probabilities are commonly very close to 0 or 1.
- Correct estimation  $\Rightarrow$  accurate prediction, but correct probability estimation is **NOT** necessary for accurate prediction (just need right ordering of probabilities)

# Naïve Bayesian Classifier: Comments

---

- Advantages :
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: ce , therefore loss of accuracy <https://eduassistpro.github.io/>
  - Practically, dependencies exist les
  - E.g., hospitals: patients: Profil history etc  
Symptoms: fever, cough etc., Disease: lung cancer, diabetes etc
  - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- Better methods?
  - Bayesian Belief Networks
  - Logistic regression / maxent

- 
- (Linear Regression and) Logistic Regression Classifier
    - See LR Slides Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

---

- Other Classification Methods

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

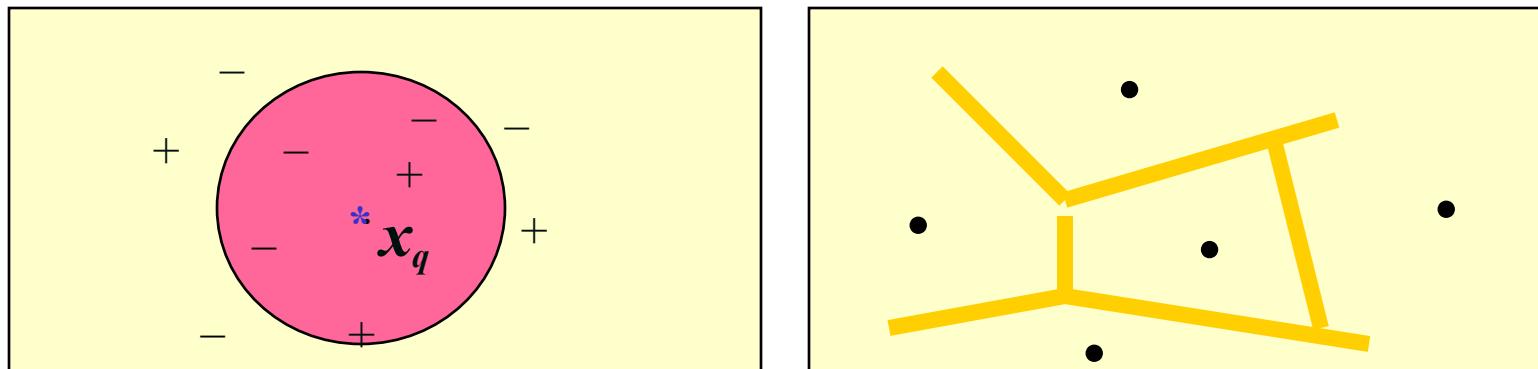
# Instance-Based Methods

---

- Instance-based learning:
  - Store training examples and delay the processing ("lazy evaluation") until a new instance must be classified
- Typical approach
  - k-nearest n
    - Instances represented in Euclidean space.

# The $k$ -Nearest Neighbor Algorithm

- All instances correspond to points in the  $n$ -D space.
- The nearest neighbor are defined in terms of Euclidean distance.
- The target function could be discrete- or real- valued.
- For discrete-valued target function <https://eduassistpro.github.io/> the most common value  $x_q$  is the  $y$  induced by the  $1$ -NN for a typical set of training examples.



# Bayesian Perspective

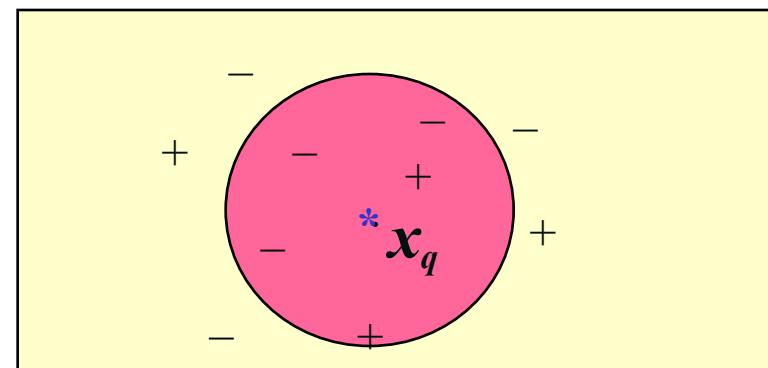
- An informal way to view kNN classifier as a Bayesian model
- Set up:
  - kNN Ball around a test instance  $\mathbf{x}$ :  $B_k(\mathbf{x})$
  - Class  $i \rightarrow$  <https://eduassistpro.github.io/>
  - Class I  $\rightarrow$   $\bigcup_i B_k(\mathbf{x}_i)$
  - Volume of  $B_k(\mathbf{x})$ : WeChat edu\_assist\_pro

$$P(C_i | \mathbf{x}) = \frac{P(\mathbf{x} | C_i)P(C_i)}{P(\mathbf{x})} \quad P(C_i) = \frac{N_i}{N}$$

$$P(\mathbf{x}) \cdot V \approx \frac{k}{N} \quad P(\mathbf{x} | C_i) \cdot V \approx \frac{k_i}{N_i}$$

estimated probability (uniform density within  $B_k(\mathbf{x})$ )

observed probability



# Effect of k

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- k acts as a smoother

# Discussion on the $k$ -NN Algorithm

- The  $k$ -NN algorithm for continuous-valued target functions
  - Calculate the mean values of the  $k$  nearest neighbors
- **Distance-weighted** nearest neighbor algorithm
  - Weight the contribution of each of the  $k$  neighbors according to their distance  $t$ 
    - giving great <https://eduassistpro.github.io/>  $w \equiv \frac{1}{d(x_q, x_i)^2}$
  - Similarly, for real-valued target  $f$
- Robust to noisy data by averaging neighbors
- Curse of dimensionality:
  - $k$ NN search becomes very expensive in high dimensional space
    - High-dimensional indexing methods, e.g., **LSH**
  - distance between neighbors could be dominated by irrelevant attributes.
    - To overcome it, axes stretch or elimination of the least relevant attributes.

# Remarks on Lazy vs. Eager Learning

---

- Instance-based learning: lazy evaluation
- Decision-tree and Bayesian classification: eager evaluation
- Key differences
  - Lazy method may consider query instance  $x_q$  when deciding how to generalize beyond it
  - Eager method constructs a chosen global approximation when seeing the query
- Efficiency: Lazy - less time training by predicting
- Accuracy
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space