INTRODUCTORY APPLIED MACHINE LEARNING

Yan-Fu Kuo

Dept. of Bio-industrial Mechatronics Engineering

National Taiwan University

Today:

- Nearest-neighbor classifiers
- Bayesian classifiers
- Logistic regression
- Ensemble methods

Outline

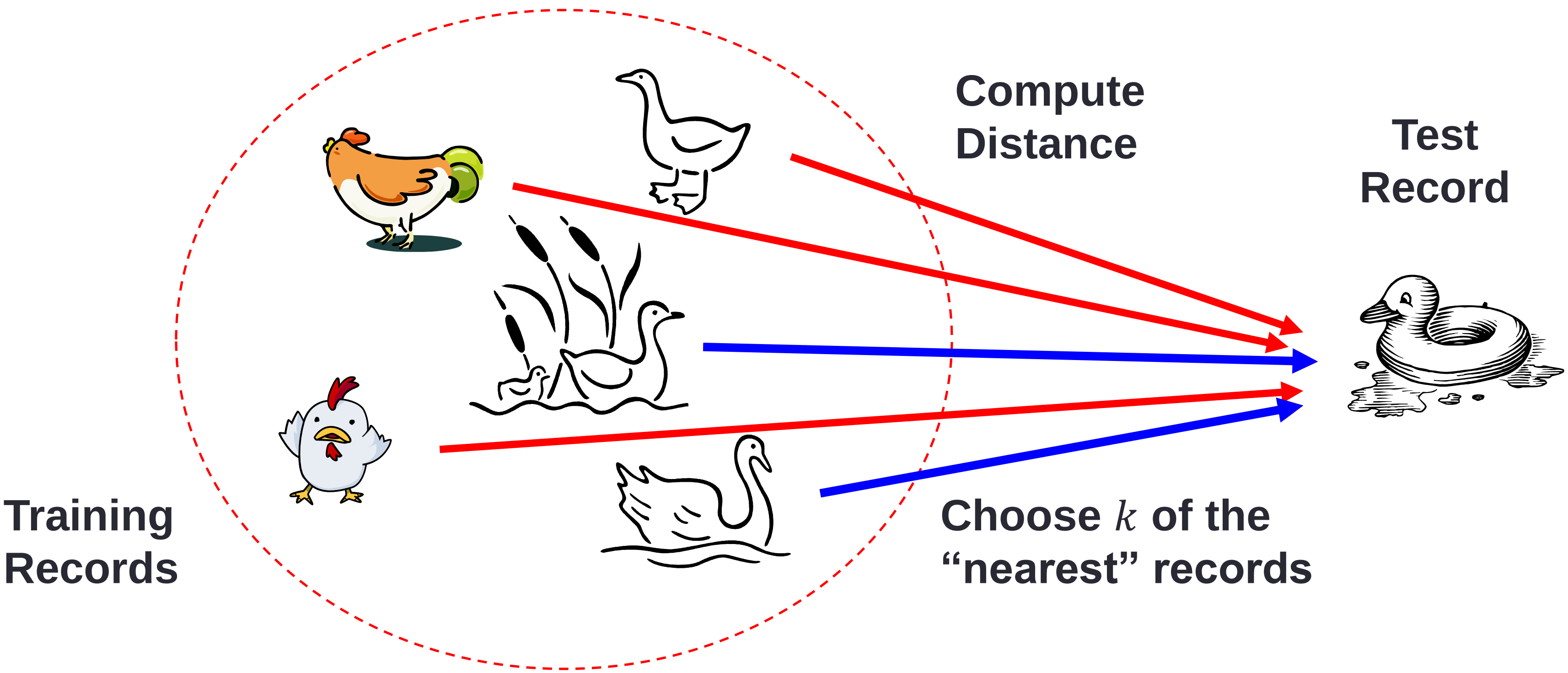
- Goal of the lecture
- K-nearest neighbor
- Naïve Bayesian classification
- Logistic regression
- Bagging
- Boosting

Goals

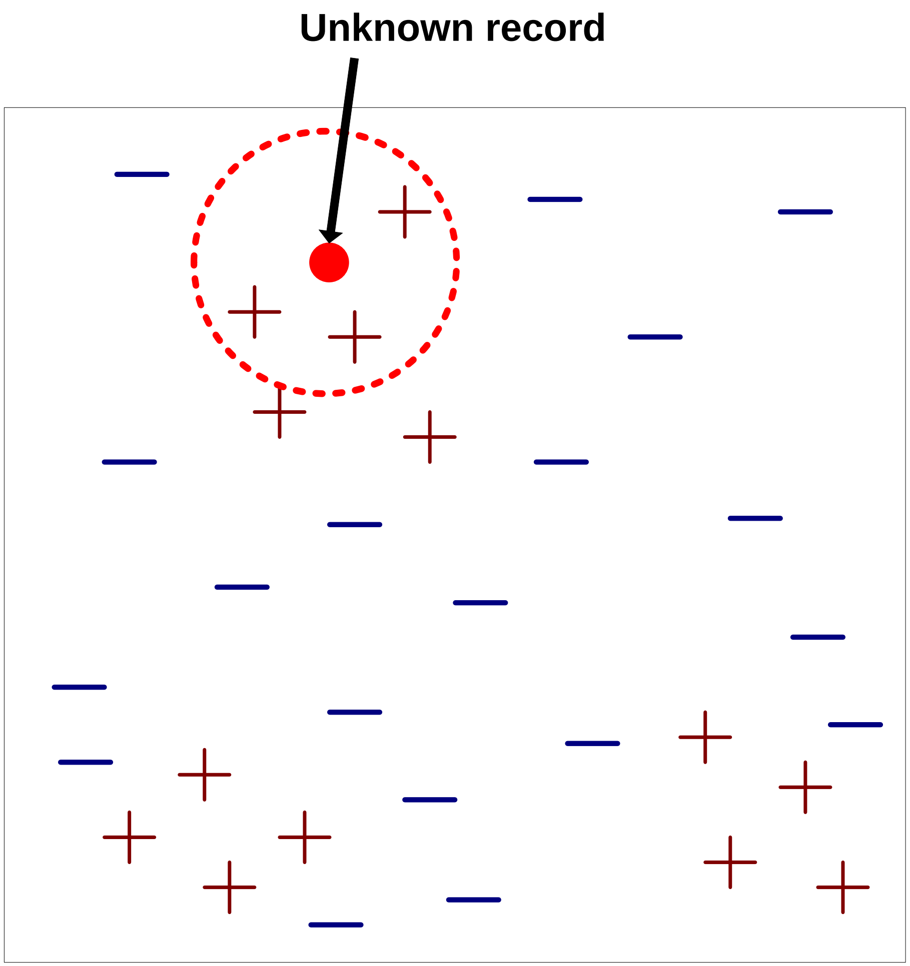
- After this, you should be able to:
  - Build k-nearest neighbor and naïve Bayesian classifiers
  - Build logistic regression models
  - Get basic ideas of ensemble methods
  - Understand the advantages and disadvantages of k-nearest neighbor, naïve Bayesian, logistic regression, and ensemble methods

k-Nearest Neighbor (kNN) Classifier

- An instance-based classifier
- Basic idea: if an animal walks like a duck, quacks like a duck, then it's probably a duck



Requirements of kNN



- Uses k "closest" points (nearest neighbors) for performing classification
- Requirements:
  1. The set of stored records
  2. Distance metric to compute distance between records
  3. The value of "k", the number of nearest neighbors to retrieve

kNN Classification Procedure

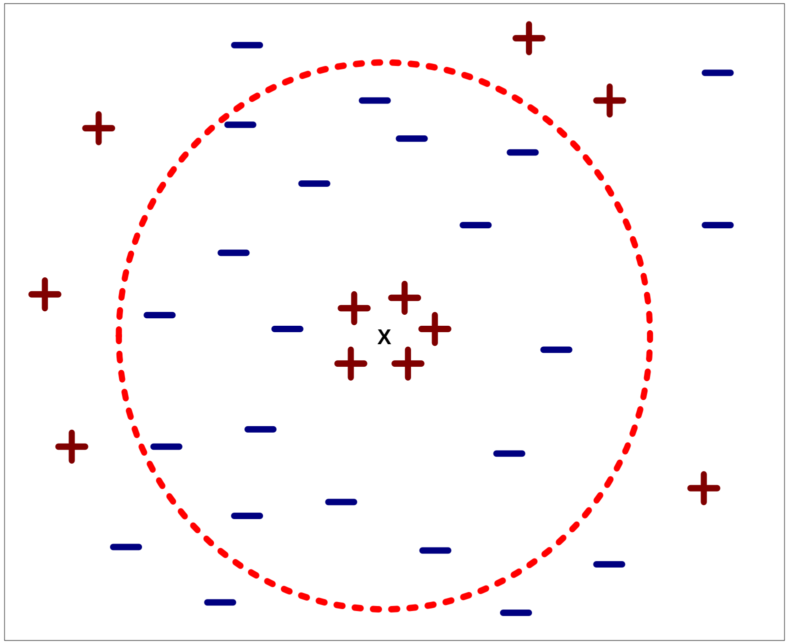
1. Compute the distance $d \in \Re$ between the unknown sample point and neighbor points

   (Typically the Euclidean (L_2) norm is used)
2. Identify k nearest neighbors
3. Take the majority vote of class labels among the k-nearest neighbors

   (Typically the weight factor $w = \frac{1}{d^2} \in \Re$ is used)

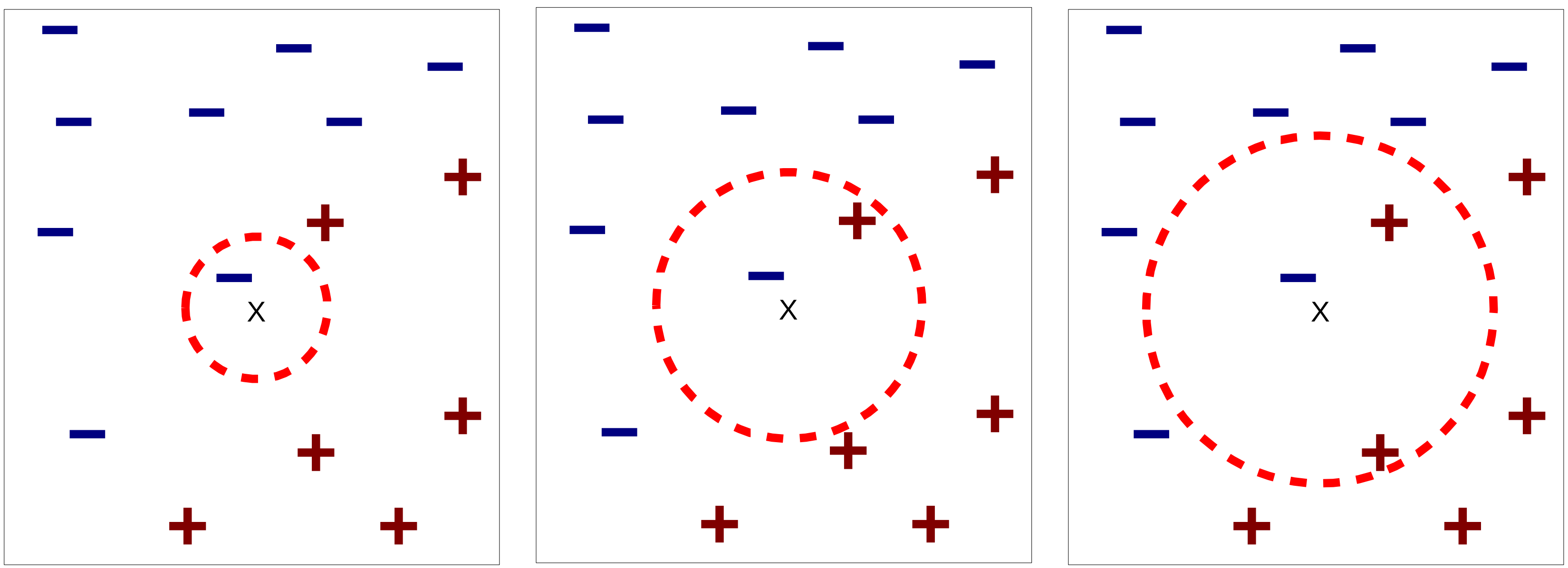
Choice of the k Value

- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other classes



The Value of “k“

- k-nearest neighbors of a record x are data points that have the k smallest distance to x

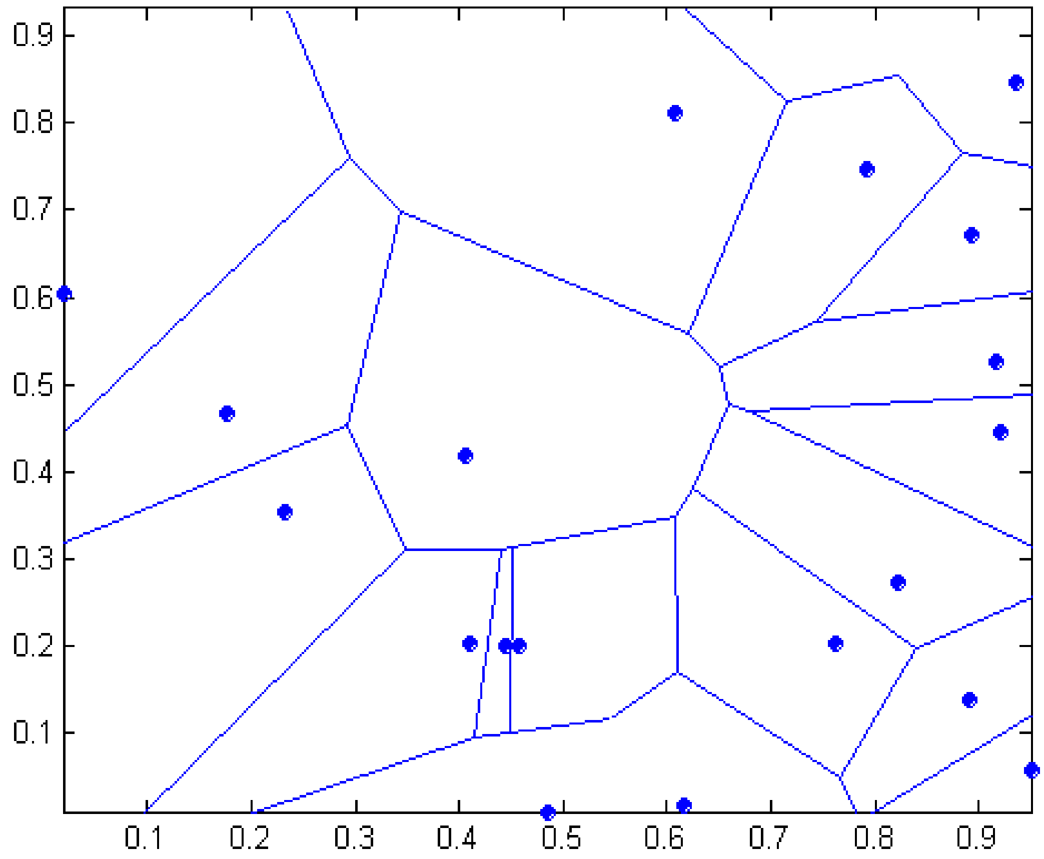


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Special Case: 1-nearest Neighbor

- Voronoi Diagram:

  decomposition of a space determined by distances to objects



Attribute Normalization

- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
- Example:
  - Height of a person may vary from 1.5m to 1.8m
  - Weight of a person may vary from 90lb to 300lb
  - Income of a person may vary from $10K to $1M

kNN Summary

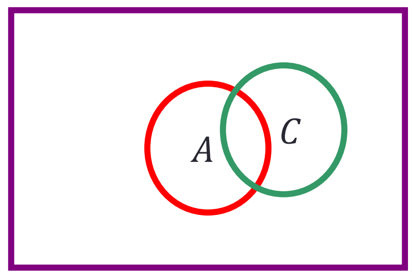
- kNN classifiers do not build models explicitly
- Classifying unknown records are relatively time consuming and computationally intensive
- Highly effective inductive inference method for noisy training data and complex target functions
- Nonparametric architecture

Bayes Theorem

- A probabilistic framework for solving classification problems
- Conditional probability:

$$P(C|A) = \frac{P(A \cap C)}{P(A)}$$

$$P(A|C) = \frac{P(A \cap C)}{P(C)}$$



- Bayes theorem:

likelihood *prior*

posterior

$$P(C|A) = \frac{P(A|C)P(C)}{P(A)}$$

evidence

Example of Bayes Theorem

- Given:
  - A doctor knows that meningitis (M) causes stiff neck (S) 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20
- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Towards Naïve Bayesian Classification

- Given a training set of attributes $\boldsymbol{x} = (x_1, x_2, \dots, x_K)$ and class y_j, $j = 1 \dots m$
- Consider each attribute and class label as a random variable
- Goal is to predict the class y_j for given $(x_1, x_2, \dots, x_K)$
- This is equivalent to find the value of y_j that maximizes the posteriori $P(y_j|\boldsymbol{x}) = P(y_j|x_1, x_2, \dots, x_K)$

Classification Using Naïve Bayes

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Evade**
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**

- Question: find the evade $y_j = Yes$ or No, given the evidence $\boldsymbol{x} = (No\ refund, Married, Inc = 120K)$
- This is equivalent to find y that maximizes $P(y_j|\boldsymbol{x}) = P(y_j|No\ refund, Married, Inc = 120K)$

Derivation of Naïve Bayes Classifier

- From Bayes theorem:

$$P(y_j|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y_j)P(y_j)}{P(\boldsymbol{x})}$$

$$\Rightarrow \; P(y_j|x_1, x_2, \dots, x_K) = \frac{P(x_1, x_2, \dots, x_K|y_j)P(y_j)}{P(x_1, x_2, \dots, x_K)} \qquad \dots\text{(a)}$$

- Note that $P(\boldsymbol{x}) = P(x_1, x_2, \dots, x_K)$ is constant for all classes
- Choosing the value of y_j that maximizes $P(y_j|\boldsymbol{x})$ is equivalent to choosing the value of y_j that maximizes $P(\boldsymbol{x}|y_j)P(y_j)$

Derivation of Naïve Bayes Classifier (Cont'd)

- Assume independence among attributes x_i, i.e.,

$$P\left(x_1, x_2, \dots, x_K \middle| y_j\right) = P\left(x_1 \middle| y_j\right) \cdot P\left(x_2 \middle| y_j\right) \cdots P\left(x_K \middle| y_j\right)$$

- Equation (a)

$$\Rightarrow P(y_j|\boldsymbol{x}) = \frac{P\left(x_1 \middle| y_j\right) \cdot P\left(x_2 \middle| y_j\right) \dots P(x_K|y_j) \cdot P(y_j)}{P(x_1, x_2, \dots, x_n)}$$

- The objective is to find the y_j that maximizes $P(y_j) \prod_{i=1}^{K} P\left(x_i \middle| y_j\right)$, i.e.,

$$y = \arg\max_{y} \left[\left(\prod_{i=1}^{K} P\left(x_i \middle| y_j = y\right) \right) P(y_j = y) \right]$$

Estimate Probabilities for Discrete Attributes

Tid	**Refund**	**Marital Status**		**Evade**
1	Yes	Single		**No**
2	No	Married		**No**
3	No	Single		**No**
4	Yes	Married		**No**
5	No	Divorced		**Yes**
6	No	Married		**No**
7	Yes	Divorced		**No**
8	No	Single		**Yes**
9	No	Married		**No**
10	No	Single		**Yes**

- Find y_j given $\boldsymbol{x} = (No\ ref, M)$

Estimate Probabilities for Continuous Attributes

- Two Common methods:
  1. Two-way split: $(x < v)$ or $(x > v)$, and choose only one of the two splits as the new attribute
  2. Probability density estimation:
     - Assume attribute follows a normal distribution
     - Use data to estimate parameters of distribution, e.g., mean and standard deviation
     - Once probability distribution is known, it can be used to estimate the conditional probability $P(x_i|y_j)$

Estimate Probabilities for Continuous Attributes

Tid	Refund	Marital Status	Taxable Income	Evade
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

- Normal distribution:

$$P(x_i|y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  for each (x_i, y_j) pair

- Example:

  Let $x_i = Income$, and $y_j = No$

  $\Rightarrow \mu_{ij} = 110$, and $\sigma_{ij}^2 = 2975$

$$P(Income = 120|No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Example

- Given that $\boldsymbol{x} = (\textit{No refund, Married, Inc} = \textit{120K})$, find the evade $y_j = Yes$ or No

- Calculate the probability:
  - $P(Refund = Yes|No) = 3/7$
  - $P(Refund = No|No) = 4/7$
  - $P(Refund = Yes|Yes) = 0$
  - $P(Refund = No|Yes) = 1$
  - $P(Marital\ Status = Single|No) = 2/7$
  - $P(Marital\ Status = Divorced|No) = 1/7$
  - $P(Marital\ Status = Married|No) = 4/7$
  - $P(Marital\ Status = Single|Yes) = 2/3$
  - $P(Marital\ Status = Divorced|Yes) = 1/3$
  - $P(Marital\ Status = Married|Yes) = 0$

- Conduct Bayes' classifier:
  - $P(\boldsymbol{x}|No) = P(No\ refund|No) \times P(Married|\ No) \times P(Inc = 120K|\ No)$ $= 4/7 \times 4/7 \times 0.0072 = 0.0024$
  - $P(\boldsymbol{x}|Yes) = P(No\ Refund|Yes) \times P(Married|Yes) \times P(Inc = 120K|\ Yes)$ $= 1 \times 0 \times 1.2 \times 10^{-9} = 0$
  - $P(\boldsymbol{x}|No)P(No) > P(\boldsymbol{x}|Yes)P(Yes)$

  $\Rightarrow$ **evade** $= No$

Another Example

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

M: mammals; N: non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

- Conditional probability:

$$P(\boldsymbol{x}|M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(\boldsymbol{x}|N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13}$$
$$= 0.0042$$

$$P(\boldsymbol{x}|M)P(M) = 0.06 \times \frac{7}{20}$$
$$= 0.021$$

$$P(\boldsymbol{x}|N)P(N) = 0.004 \times \frac{13}{20}$$
$$= 0.0027$$

$$P(\boldsymbol{x}|M)P(M) > P(\boldsymbol{x}|N)P(N)$$
$$\Rightarrow Mammals$$

Avoiding the Zero-probability Problem

- Naïve Bayesian prediction requires each conditional probability be non-zero; otherwise, the predicted probability will be zero

$$P(\boldsymbol{x}|y_j) = \prod_{i=1}^{K} P(x_i|y_j)$$

- Corrected probability are used

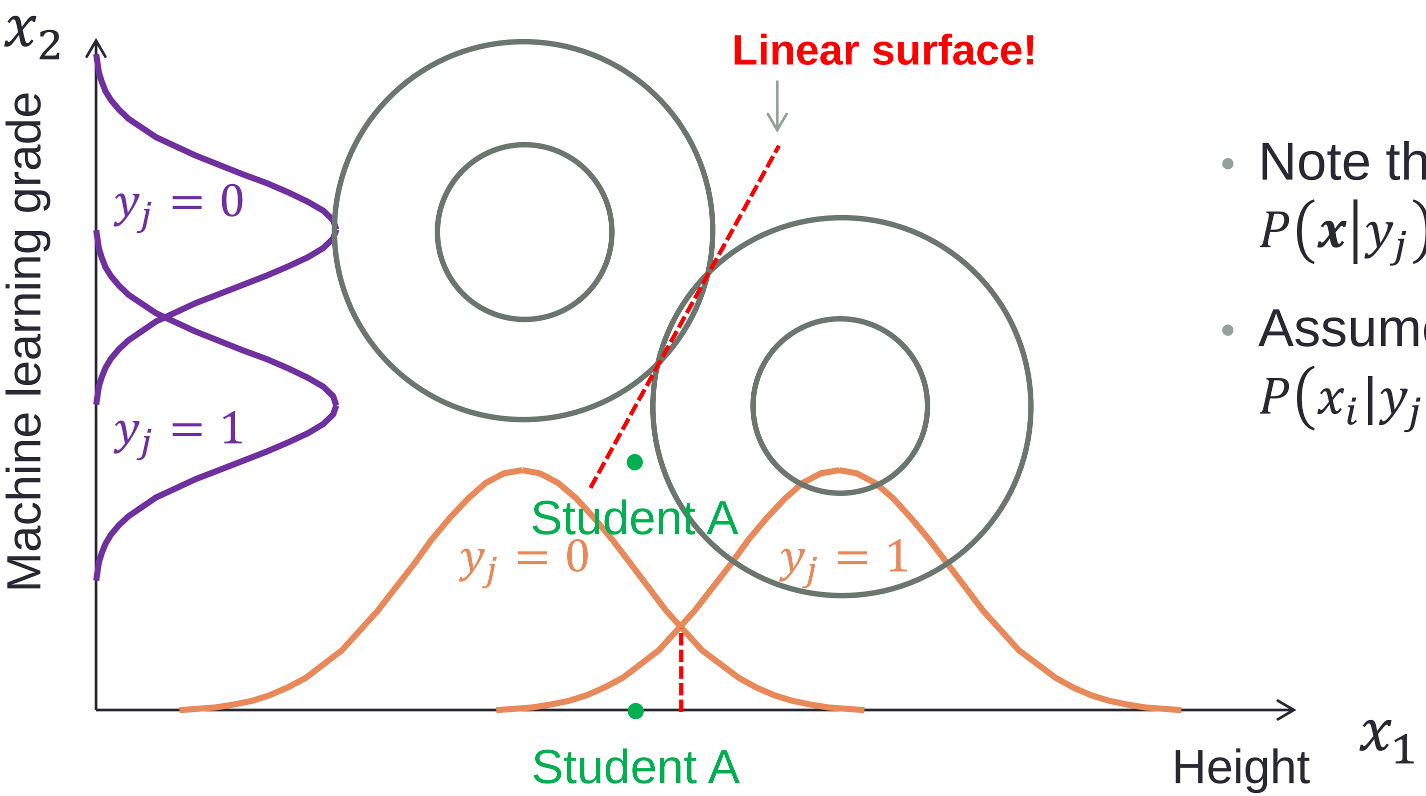
$$\text{Laplace: } P(x_i|y_j) = \frac{N_{ij}+1}{N_j+|y|}$$

$$\text{m-estimate: } P(x_i|y_j) = \frac{N_{ij}+mp}{N_j+m}$$

where $|y|$ is number of classes, p is a predetermined parameter, and m is the equivalent sample size

Geometric Interpretation of Naïve Bayes

- Consider boolean y_j, x_i normally distributed, and $P(y_j = 1) = 0.5$
- Naïve Bayes: $y = \arg\max_y P(y_j = y) \prod_{i=1}^{K} P(x_i | y_j = y)$



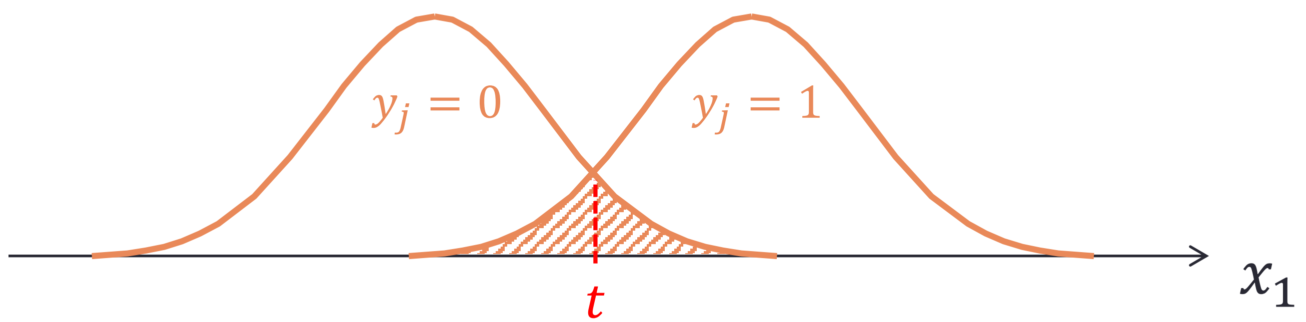
- Note that $P(\boldsymbol{x}|y_j) = \prod_{i=1}^{K} P(x_i|y_j)$
- Assume $P(x_i|y_j) \sim N(\mu_{ij}, \sigma)$

The Minimum Possible Error

- Conditional independence assumption is satisfied
- Assume that we know $P(x_i|y_j)$, and $P(y_j = 1) = 0.5$

$$P(err) = P(\text{pred } y_j = 1 \text{ but } y_j = 0) + P(\text{pred } y_j = 0 \text{ but } y_j = 1)$$

$$= \int_{-\infty}^{t} P(x_1|\, y_j = 1)\, P(\, y_j = 1) + \int_{t}^{\infty} P(x_1|\, y_j = 0)\, P(\, y_j = 0)$$



Naïve Bayes Summary

- Assumption of independently continuous distribution may not hold for some attributes
- Easy to implement
- Robust to isolated noise points
- Can handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes

Logistic Regression Problem Definition

- Objective: estimate $P(y_j|\boldsymbol{x}) = f(\boldsymbol{x})$ for given $\boldsymbol{x} \in \Re^K$
- Strategy: follow naïve Bayes rule
- Assumptions:
  - y is Boolean (i.e., $y = 1$ or 0)
  - $P(y = 1) = \gamma$ and $P(y = 0) = 1 - \gamma$
  - All x_i are conditionally independent for given y
  - $P(x_i|y_j) \sim N(\mu_{ij}, \sigma_i)$, i.e., Gaussian distributed

Logistic Regression Derivation

- Bayes rule indicates that

$$\boxed{P(y=1|\boldsymbol{x})} = \frac{P(\boldsymbol{x}|y=1)P(y=1)}{P(\boldsymbol{x}|y=1)P(y=1)+P(\boldsymbol{x}|y=0)P(y=0)}$$

$$= \frac{1}{1+\frac{P(\boldsymbol{x}|y=0)P(y=0)}{P(\boldsymbol{x}|y=1)P(y=1)}} = \frac{1}{1+\exp(\ln\left(\frac{P(\boldsymbol{x}|y=0)P(y=0)}{P(\boldsymbol{x}|y=1)P(y=1)}\right))}$$

$$= \frac{1}{1+\exp(\ln\left(\frac{P(y=0)}{P(y=1)}\right)+\ln\left(\prod_i \frac{P(x_i|y=0)}{P(x_i|y=1)}\right))}$$

$$= \frac{1}{1+\exp(\ln\left(\frac{1-\gamma}{\gamma}\right)+\boxed{\sum_i \ln\left(\frac{P(x_i|y=0)}{P(x_i|y=1)}\right)})}$$

Logistic Regression Derivation (Cont'd)

$$\boxed{\sum_i \ln\left(\frac{P(x_i|y=0)}{P(x_i|y=1)}\right)} = \sum_i \ln\left(\frac{\frac{1}{\sqrt{2\pi\sigma_i^2}}\exp(\frac{-(x_i-\mu_{i0})^2}{2\sigma_i^2})}{\frac{1}{\sqrt{2\pi\sigma_i^2}}\exp(\frac{-(x_i-\mu_{i1})^2}{2\sigma_i^2})}\right)$$

$$= \sum_i \ln\left(\exp\left(\frac{(x_i-\mu_{i1})^2-(x_i-\mu_{i0})^2}{2\sigma_i^2}\right)\right)$$

$$= \sum_i \frac{\left(x_i^2-2x_i\mu_{i1}+\mu_{i1}^2\right)-\left(x_i^2-2x_i\mu_{i0}+\mu_{i0}^2\right)}{2\sigma_i^2}$$

$$= \sum_i \left(\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}x_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)$$

Logistic Regression Derivation (Cont'd)

$$\boxed{P(y=1|\boldsymbol{x})} = \frac{1}{1+\exp\left(\ln\left(\frac{1-\gamma}{\gamma}\right) + \boxed{\sum_i \left(\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}x_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)}\right)}$$

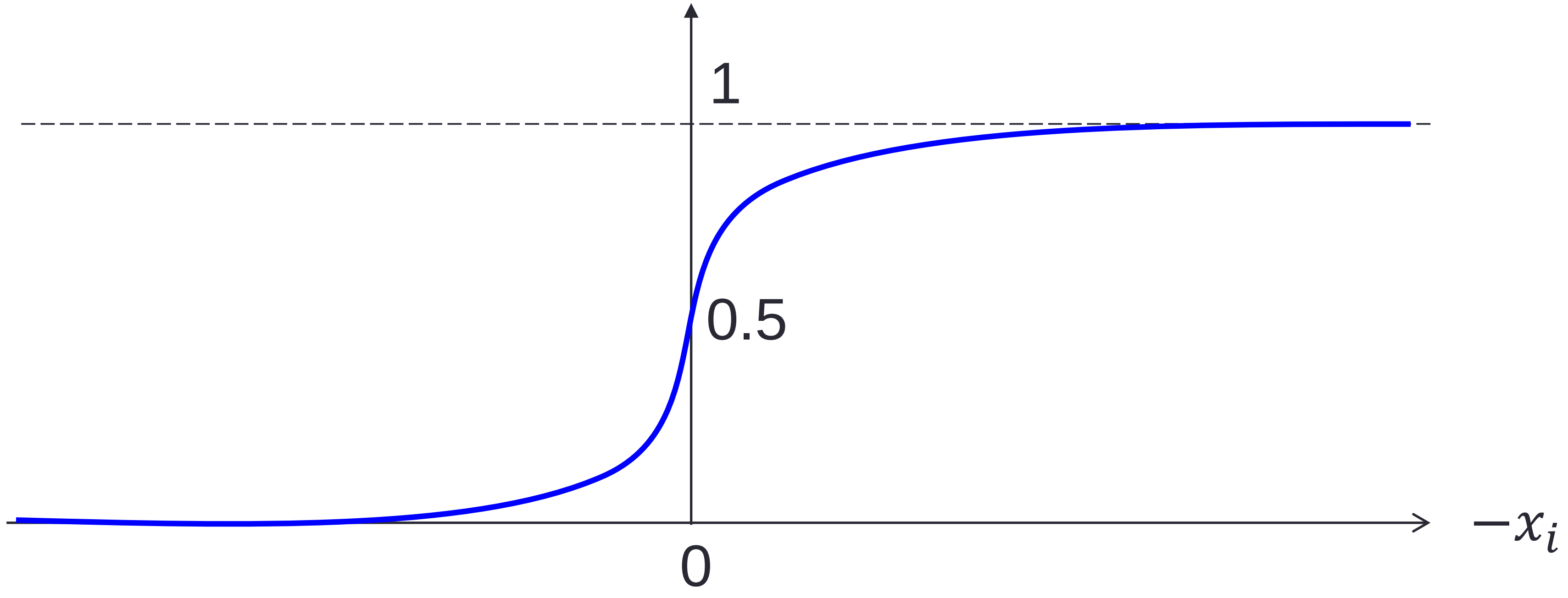
$$= \frac{1}{1+\exp\left(w_0 + \sum_{i=1}^{K} w_i x_i\right)} \quad \Leftarrow \textit{A sigmoid equation!}$$

where $w_0 = \ln\left(\frac{1-\gamma}{\gamma}\right) + \sum_i \left(\frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)$, $\quad w_i = \frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}$

$$\Rightarrow \quad \underline{P(y=0|\boldsymbol{x})} = 1 - \boxed{P(y=1|\boldsymbol{x})} = \frac{\exp\left(w_0 + \sum_{i=1}^{K} w_i x_i\right)}{1+\exp\left(w_0 + \sum_{i=1}^{K} w_i x_i\right)}$$

Logistic Function

$$P(y=1|\boldsymbol{x}) = \frac{1}{1+\exp(\sum_{i=1}^{K} w_i x_i)}$$



Logistic Regression Derivation (Cont'd)

- This indicates:

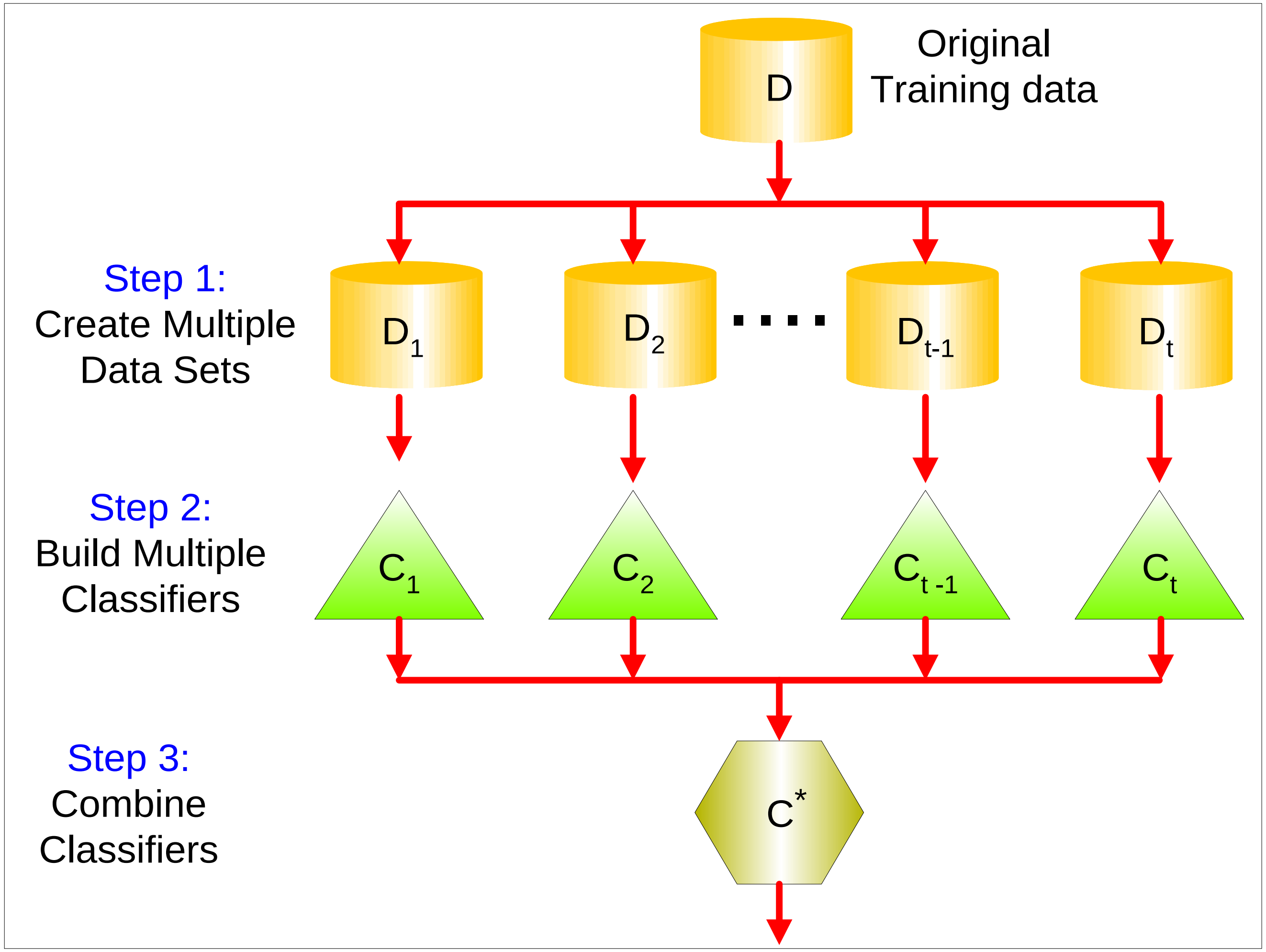
$$\frac{P(y=0|\boldsymbol{x})}{P(y=1|\boldsymbol{x})} = \exp\left(w_0 + \sum_{i=1}^{K} w_i x_i\right)$$

which implies

$$\ln\left(\frac{P(y=0|\boldsymbol{x})}{P(y=1|\boldsymbol{x})}\right) = w_0 + \sum_{i=1}^{K} w_i x_i$$

Ensemble Methods

- General idea: combine multiple classifiers



Why Does It Work?

- Suppose there are 25 “base” classifiers
- Each classifier has an error rate $\varepsilon = 0.35$
- Assume classifiers are independent
- Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i}(1-\varepsilon)^{25-i} = 0.06$$

- The ensemble makes a wrong prediction only if more than half of the base classifiers predict incorrectly

Typical Ensemble Methods

- Bagging (by Leo Breiman):

  Resampling, i.e., generating new training samples from the original sample set, based on uniform distribution

- Boosting:

  Adaptively changes the weights of samples in resampling to tackle those "hard to classify" samples

Bagging

- Sample with replacement from the original data set according to a uniform probability distribution
- Examples chosen during each bagging:

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bagging sample set
- A particular training data has a probability of $1 - 1/N$ of not being picked, where N is number of samples
- A sample has probability $1 - (1 - 1/N)^N$ of being selected
- The probability is equal to 0.632 if $N \to \infty$, so this method is also called 0.632 bootstrap

Boosting

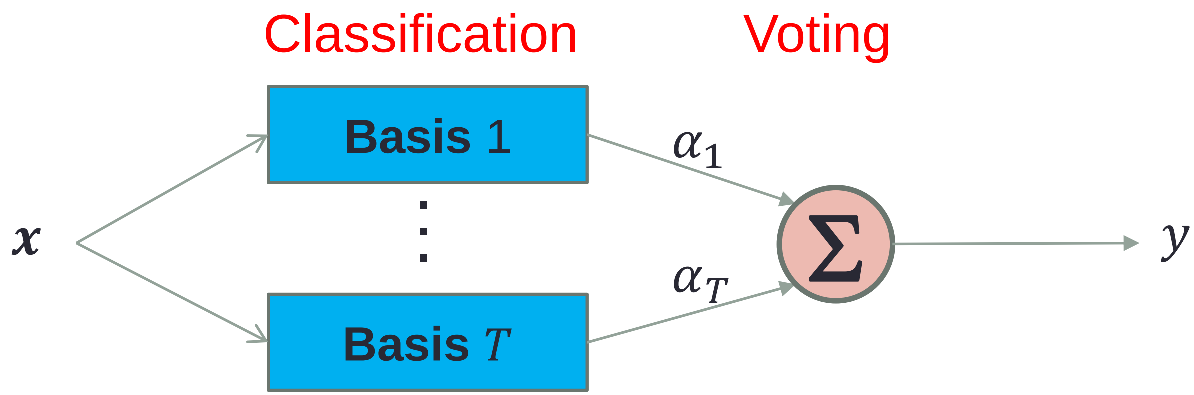
- Sample with replacement from the original data set
- Iteratively change distribution of training data by focusing more on previously misclassified records
- Initially, all n records are assigned equal weights
- Records that are wrongly classified will have their weights increased in the future iteration

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

Example 4 is hard to classify

Adaptive Boosting (AdaBoost) Classifier

- Suppose there exists T "basis" classifiers C_t, $t = 1 \dots T$
- Each classifier is associated with a weight α_t
- For a query input $\boldsymbol{x}$, the output y is determined by weighted majority voting:



AdaBoost Classifier

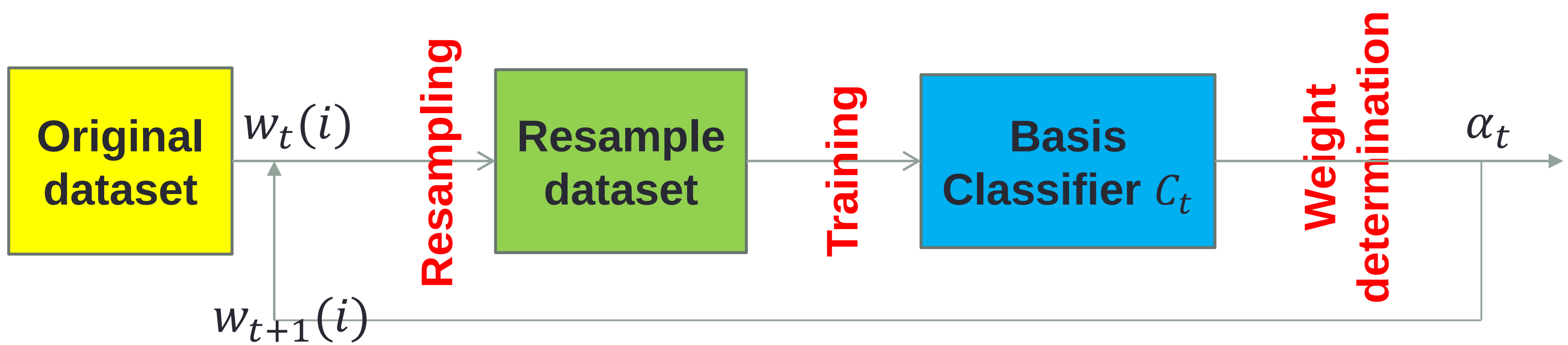
- Output is determined by weighted majority voting:

$$C^*(\boldsymbol{x}) = \arg\max_{y} \sum_{t=1}^{T} \alpha_t I(C_t(\boldsymbol{x}) = y)$$

$$\text{where } \begin{cases} I(p) = 1 & \text{when } p \text{ is true} \\ I(p) = 0 & \text{otherwise} \end{cases}$$

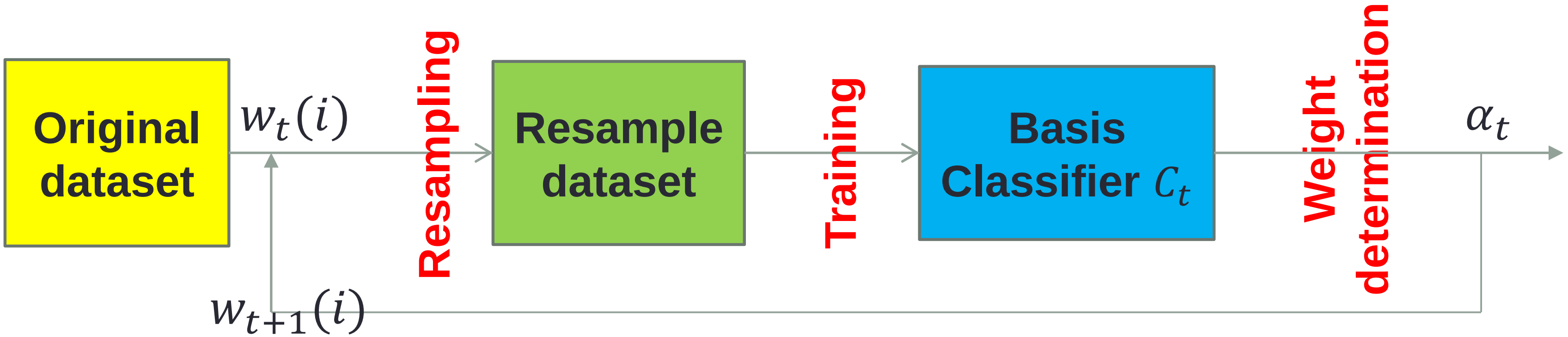
Basis Classifier Training

- Let $\{(\boldsymbol{x}_i, y_i)|i = 1 \dots N\}$ denote a set of samples, $y_i = \{+1, -1\}$
- Objective: generate basis classifiers $C_t, t = 1 \dots T$
- The basis classifiers are developed iteratively
- In each iteration, data points are sampled with replacement using weight $w_t(i)$
- The initial sample weights $w_1(i) = \frac{1}{N}, i = 1 \dots N$



Basis Classifier Training Steps

- Determine the follows in each iteration
  1. The error rate ε_t for the basis classifier C_t
  2. The weights α_t for the basis classifier C_t
  3. The resampling weights w_{t+1} for the next iteration



Basis Classifier Error Rate ε_t

- The (misclassification) error rate of a basis classifier C_t is:

$$\varepsilon_t = \frac{1}{N}\sum_{i=1}^{N} w_t(i) I(C_t(\boldsymbol{x}_i) \neq y_i)$$

where $w_t(i) \in \Re$ is the weight assigned to sample $(\boldsymbol{x}_i, y_i)$, and

$$\begin{cases} I(p) = 1 & \text{when } p \text{ is true} \\ I(p) = 0 & \text{otherwise} \end{cases},$$

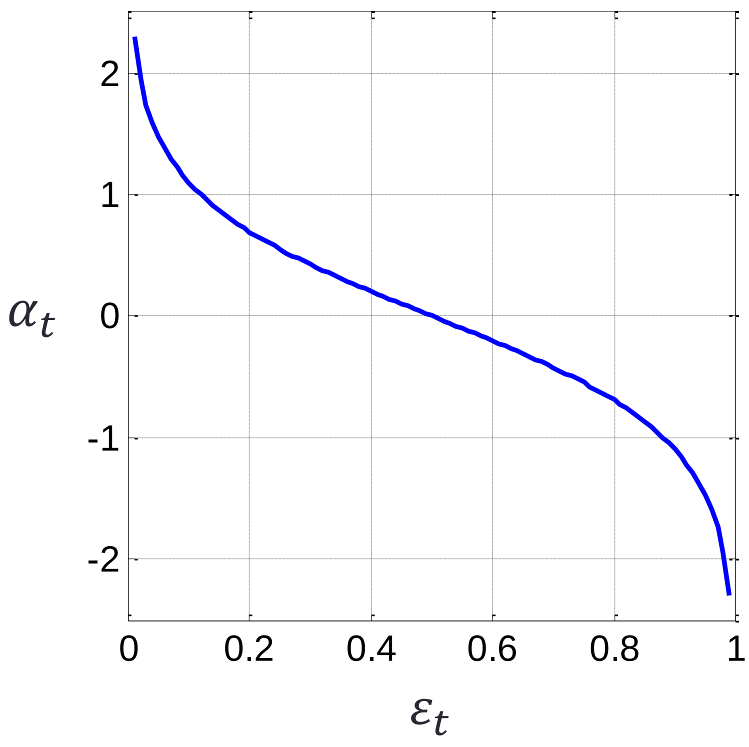
- Note that $0 \leq \varepsilon_t \leq 1$

Basis Classifier Weight α_t

- The weight $\alpha_t \in \Re$ of a basis classifier C_t is defined as

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

- The lower a base classifier's error rate ε_t, the higher its weight α_t for voting



Resampling Sample Weights $w_{t+1}(i)$

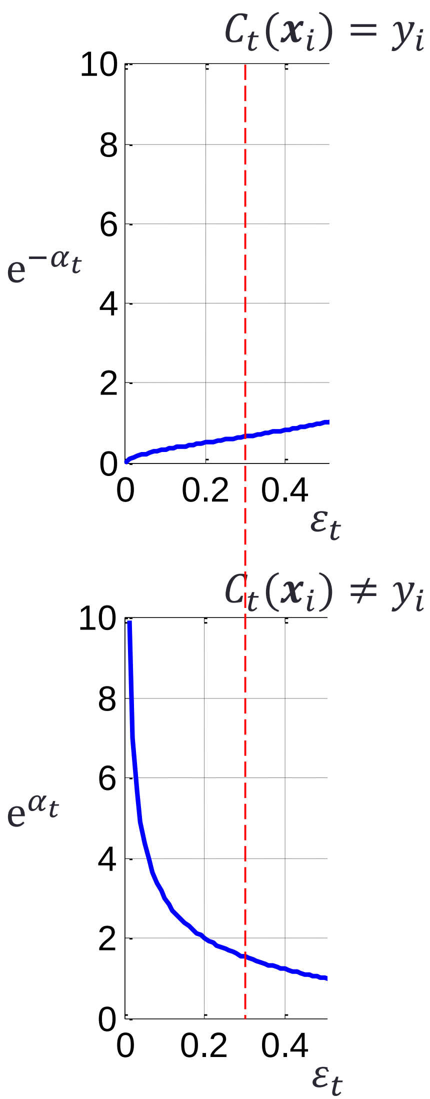
- The weights of sample $(\boldsymbol{x}_i, y_i)$ for next iteration is

$$w_{t+1}(\mathrm{i}) = \frac{w_t(i)}{z_t}\begin{cases} \mathrm{e}^{-\alpha_t} & \text{if } C_t(\boldsymbol{x}_i) = y_i \\ \mathrm{e}^{\alpha_t} & \text{if } C_t(\boldsymbol{x}_i) \neq y_i \end{cases}$$

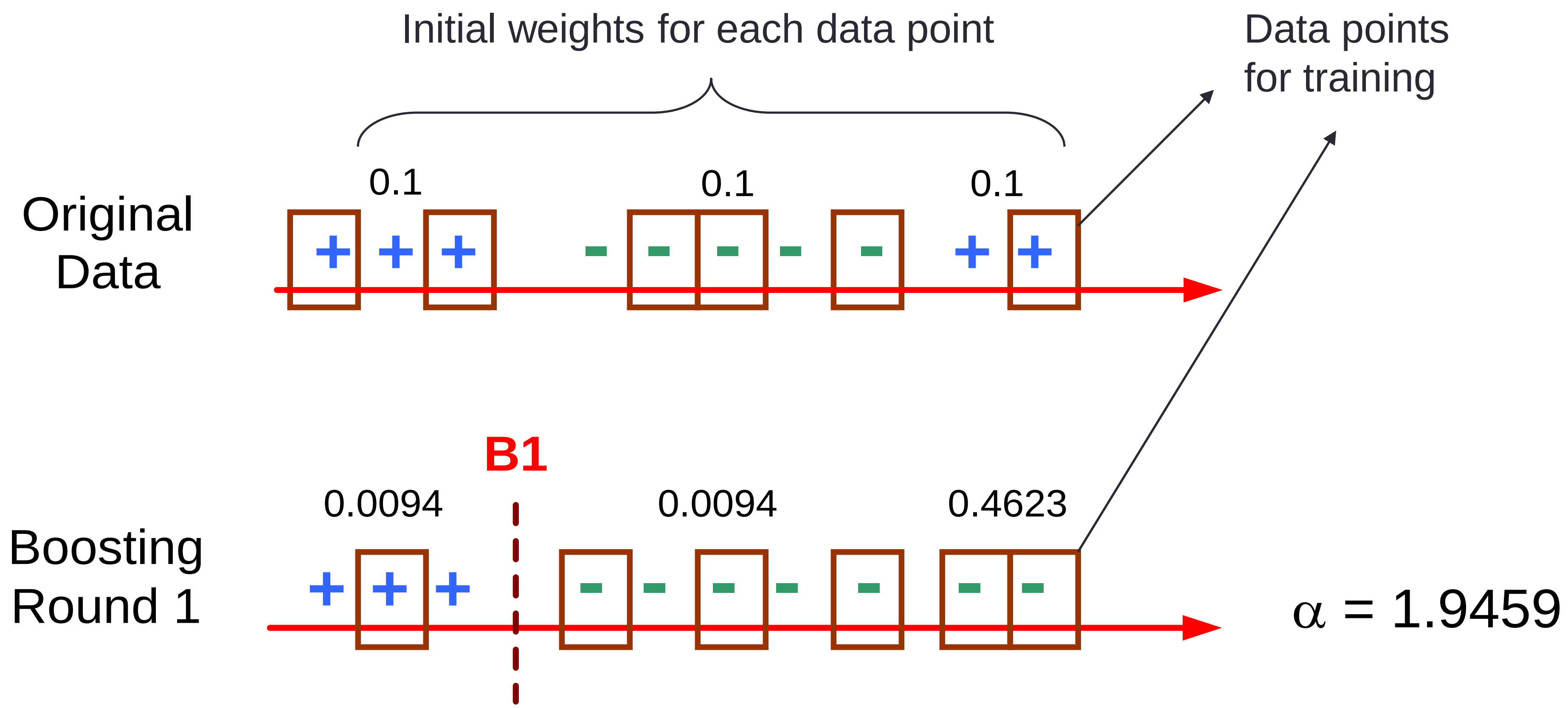
where z_t is a normalization factor that ensures

$$\sum_i w_{t+1}(i) = 1$$

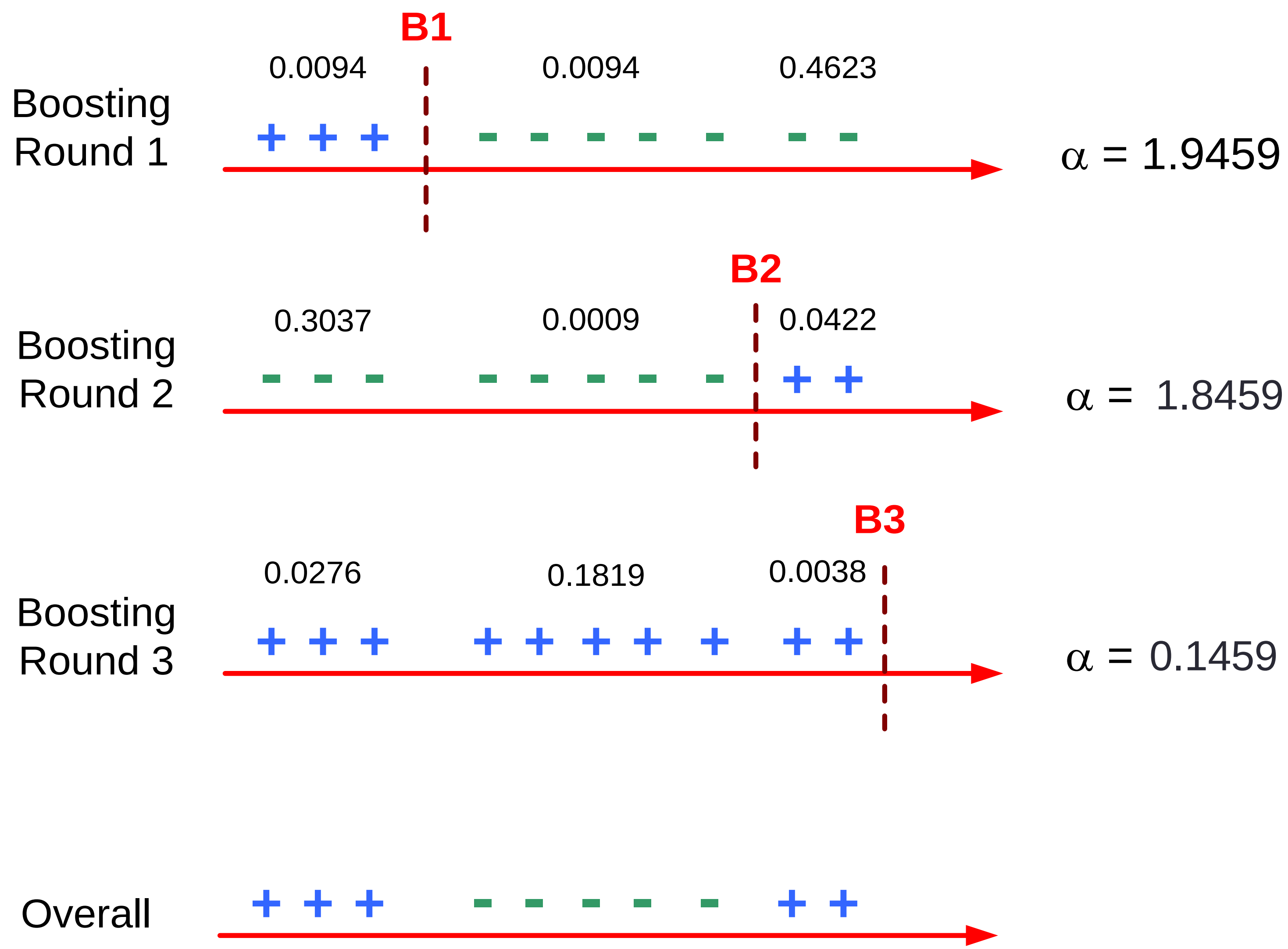
- The weights of incorrectly classified samples is increased



Illustrating AdaBoost



Illustrating AdaBoost



Ensemble Method Summary

- Ensemble methods use multiple models to obtain better predictive performance
- Bagging increases prediction accuracy because it reduces the variance of the individual classifier

Acknowledgement

- Especially thank Dr. Tom Mitchell for sharing his valuable teaching material in this course

References

- P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*