



Artificial Intelligence III:
Artificial Intelligence and Deep Learning

Lecture 5 Ensemble

Dr. Patrick Chan
patrickchan@ieee.org
South China University of Technology, China



Agenda

- ◆ Why Ensemble
- ◆ Fusion
- ◆ Diversity
- ◆ Construction Method



2

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5



Why Ensemble?

- ◆ How to **choose the best model** for a classification problem?
 - Trial and Error
 - Train many classifiers with different settings
 - Evaluate how good they are

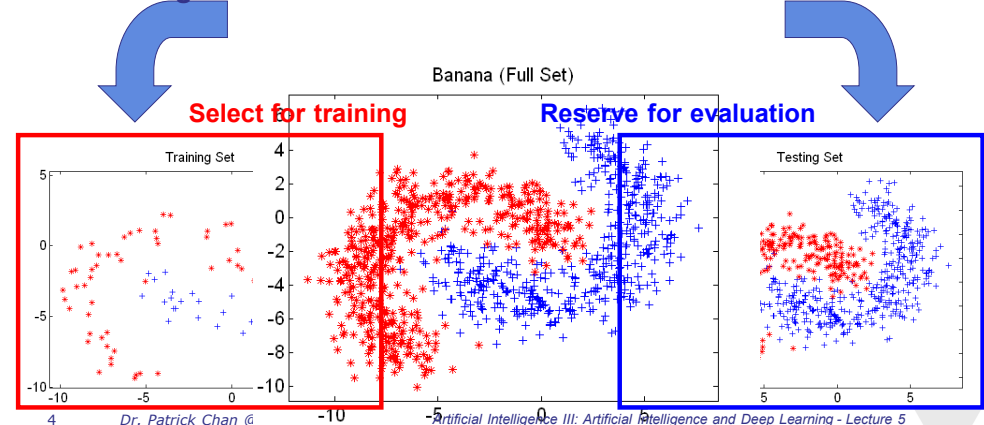


Why Ensemble?

- ◆ Banana Artificial Dataset
 - 2 class problem
 - 2 features and 1000 samples

10% Training

90% Testing



3

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5

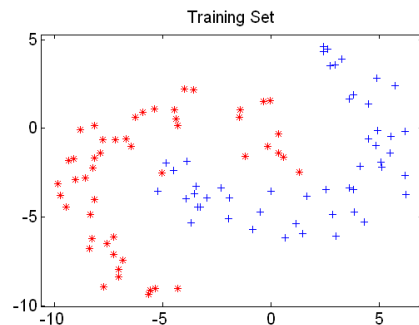
Dr. Patrick Chan @

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5

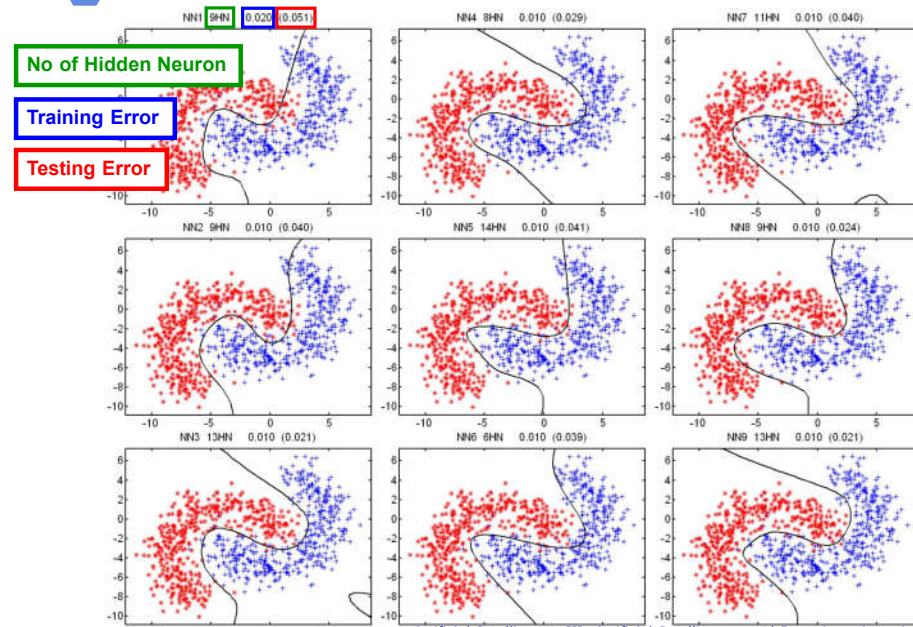


Why Ensemble?

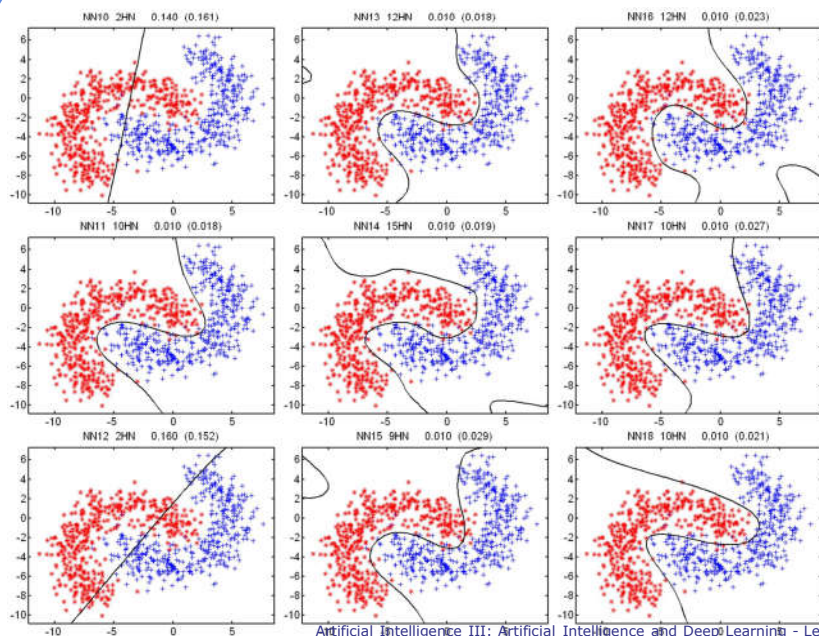
- ◆ Assume a simple MLPNN with one hidden layer is used
- ◆ No idea how many hidden neurons should be used
- ◆ Many 3-layer MLPNNs with different settings are trained



Why Ensemble?



Why Ensemble?



Why Ensemble?

- ◆ How to choose the best classifier?

Selection Criteria

Training Accuracy

- Many choices
- #2, #3..

Training Accuracy+Complexity

- Classifier with smallest number of hidden neurons and lowest training accuracy?
- #6?

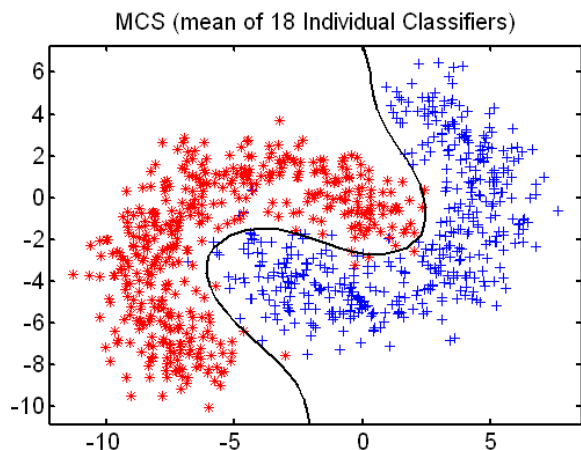
		Training Error	Test Error
#	HN		
1	9	0.020	0.051
2	9	0.010	0.040
3	13	0.010	0.021
4	8	0.010	0.029
5	14	0.010	0.041
6	6	0.010	0.039
7	11	0.010	0.040
8	9	0.010	0.024
9	13	0.010	0.021
10	2	0.140	0.161
11	10	0.010	0.018
12	2	0.160	0.152
13	12	0.010	0.018
14	15	0.010	0.019
15	9	0.010	0.029
16	12	0.010	0.023
17	10	0.010	0.027
18	10	0.010	0.021

Which criterion is the best?



Why Ensemble?

- ◆ How about combine all of them?



Training Error = 0.0100
Testing Error = 0.0167

Its performance is better than the best individual classifier (0.018)

But no guarantee!

9

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5



Why Ensemble?

- ◆ **Drawbacks of selecting the BEST**
 - Selecting a wrong one definitely leads to erroneous result
 - The "best" classifier is not necessarily the ideal choice
 - Different classifiers may contain different valuable information
 - Potentially valuable information may be lost by discarding results of less-successful classifiers
 - A single classifier may not be adequate to handle today's increasingly complex problems

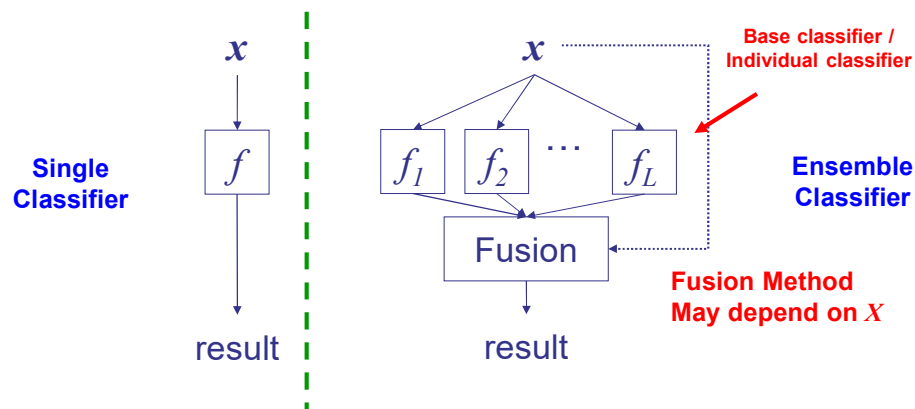
10

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5



Ensemble



- ◆ Sometimes called **Multiple Classifier System (MCS)**
- ◆ Consists of a set of individual classifiers united by a fusion method

11

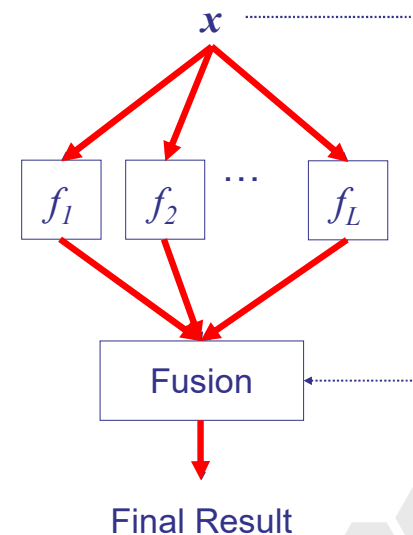
Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5



Ensemble

1. Sample x is fed into each base classifier
2. Each base classifier makes it's own decision
3. Final decision is made by combining all individual decisions



12

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5



Ensemble

- ◆ **Ensemble must be better than Single?**
 - All cases: NO!
 - But practically, yes for many cases

13

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5



Ensemble

- ◆ **Three factors** affecting the performance (accuracy) of ensemble:
 - **Accuracy of base classifiers**
 - How good are the base classifiers?
 - **Fusion Method**
 - How to combine classifiers?
 - **Diversity among base classifiers**
 - How different are the decisions reached by the classifiers?

14

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5



Three Key Factors

Accuracy of Base Classifiers

- ◆ **Performance of a base classifier** is affected by
 - **Training Dataset** (sample and feature)
 - **Learning Model** (type of classifier)
 - **Parameters** (e.g. neuron and layer # in a NN)
- ◆ **If base classifiers are poor, ensemble cannot be good**
 - But we still can hope it will be better than base classifiers

15

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5



Three Key Factors

Fusion Method

- ◆ A method to arrive at a **group decision**
- ◆ Two categories based on classifier output:
 - **Label output**
 - Output is a **class ID**
 - E.g. $[1 \ 0 \ 0]$
x is Class 1



	y1	y2	y3
Classifier 1	1	0	0
Classifier 2	0	1	0
Classifier 3	1	0	0
Classifier 4	0	0	1

$$\Rightarrow \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,c} \\ y_{2,1} & y_{2,2} & & y_{2,c} \\ \vdots & & \ddots & \vdots \\ y_{L,1} & y_{L,2} & \cdots & y_{L,c} \end{bmatrix} = D(x)$$

16

Dr. Patrick Chan @ SCUT

Artificial Intelligence III: Artificial Intelligence and Deep Learning - Lecture 5





Fusion Method

- ◆ A method to arrive at a **group decision**
- ◆ Two categories based on classifier output:

- Continuous-valued output

- Output a **real value** (probability) for each class

- E.g. [0.7 0.1 0.2]

x is Class 1 is 0.7,
Class 2 is 0.1,
Class 3 is 0.2



	y1	y2	y3
Classifier 1	0.7	0.1	0.2
Classifier 2	0.2	0.3	0.5
Classifier 3	0.8	0.1	0.1
Classifier 4	0.7	0.3	0.0

$$\begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,c} \\ y_{2,1} & y_{2,2} & & y_{2,c} \\ \vdots & & \ddots & \vdots \\ y_{L,1} & y_{L,2} & \cdots & y_{L,c} \end{bmatrix} = D(x)$$



Decision Profile

- ◆ **Decision Profile (D)** of a number of base classifier f_i ($i = 1 \dots L$)

Row: Outputs of a base classifier on all classes

$$D(x) = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,c} \\ y_{2,1} & y_{2,2} & & y_{2,c} \\ \vdots & & \ddots & \vdots \\ y_{L,1} & y_{L,2} & \cdots & y_{L,c} \end{bmatrix}$$

Column: Outputs of all base classifier on a class

- c : the number of classes
- L : the number of base classifiers
- y_{ij} : the output of i th classifier on j th class



Label Output

- ◆ **Label output** of a base classifier can be represented by **one-hot**

- 1 indicate the **class x belongs to**
- Other classes are 0

- ◆ For Example:

- 3-class problem
- 4 base classifiers and their decisions are class 2, class 3, class 1 and class 2

Row: a classifier
Column: a class

$$D(x) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



Label Output

- ◆ Majority Vote



- Also called the **Plurality**

- The **class** with the **most votes** $\max_{j=1 \dots c} \sum_{i=1}^L d_{i,j}$

- For example:

$$D(x) = \begin{bmatrix} y_1 & y_2 & y_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Row: a classifier
Column: a class

Class 2 is the majority



Label Output

◆ Simple Majority



- A class has **50% + 1 votes** ($\max_{j=1\dots c} \sum_{i=1}^L d_{i,j} > L/2$)
- More strictive than Majority Vote
- Many unknown cases

$$D(x) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

No decision

◆ Unanimity



- All base classifiers **have the same decision**
- $\max_{j=1\dots c} \sum_{i=1}^L d_{i,j} = L$
- Many unknown cases

$$D(x) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

No decision



Label Output

- Voting method **assumes** each base classifier has **same classification ability**
- However, in most cases, this is not true
- Weighted Majority Vote**
 - Assign a **weight (w_i)** to the **i th base classifier** based on its **ability**
 - A **large w** indicates **more accurate**
 - E.g. Evaluated by accuracy on Training Set
 - The class is y_k if $\sum_{i=1}^L w_i d_{i,k} = \max_{j=1\dots c} \sum_{i=1}^L w_i d_{i,j}$



Label Output

◆ Example: 3 classes, 5 base classifiers

$$D(x) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Unanimity ? (all votes)

Simple Majority y_1 (votes > 50%)Majority Vote y_1 (most votes)

$$w = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.2 \\ 0.5 \\ 0.1 \end{bmatrix}$$

Weighted Majority Vote y_2

Class 1 $0.1 \times 1 + 0.1 \times 1 + 0.2 \times 1 + 0.5 \times 0 + 0.1 \times 0 = 0.4$
 Class 2 $0.1 \times 0 + 0.1 \times 0 + 0.2 \times 0 + 0.5 \times 1 + 0.1 \times 0 = 0.5$
 Class 3 $0.1 \times 0 + 0.1 \times 0 + 0.2 \times 0 + 0.5 \times 0 + 0.1 \times 1 = 0.1$



Continuous-valued Output

- Base classifier **outputs a real value** (not a label) for each class
- The values in D are a real number
- For Example:
 - 3-class problem
 - 4 base classifiers
- Based on D , **fusion** of Continuous-valued Output **calculated a real value for each class** ($\mu_j, j = 1..c$)

Row: a classifier
 Column: a class

$$D(x) = \begin{bmatrix} 0.1 & 0.7 & 0.2 \\ 0.0 & 0.2 & 0.8 \\ 0.9 & 0.1 & 0.0 \\ 0.1 & 0.8 & 0.1 \end{bmatrix}$$



Continuous-valued Output

◆ Statistical Operator

■ Product

$$\mu_j(x) = \prod_{i=1}^L d_{i,j}(x)$$

■ Minimum

$$\mu_j(x) = \min_i \{d_{i,j}(x)\}$$

■ Simple Mean

$$\mu_j(x) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(x)$$

■ Median

$$\mu_j(x) = \text{median}_i \{d_{i,j}(x)\}$$

■ Maximum

$$\mu_j(x) = \max_i \{d_{i,j}(x)\}$$

■ Trimmed Mean

- Values are sorted and K percent of the values are dropped on each side
- Find the mean of remaining values



Continuous-valued Output

◆ Weighted Average

■ L Weights

- One **weight** per base **classifier**

$$\mu_j(x) = \sum_{i=1}^L w_i d_{i,j}(x)$$

■ c × L Weights

- Weights** are specific for each **class** per base **classifier**

$$\mu_j(x) = \sum_{i=1}^L w_{ij} d_{i,j}(x)$$



Continuous-valued Output

◆ Example:

- 3 classes
- 5 base classifiers

$$D(x) = \begin{bmatrix} 0.6 & 0.4 & 0.1 \\ 0.7 & 0.2 & 0.7 \\ 0.5 & 0.2 & 0.1 \\ 0.5 & 0.7 & 0.6 \\ 0.5 & 0.8 & 0.6 \end{bmatrix}$$

Product	0.05	0.01	0.00	y ₁
Median	0.5	0.4	0.6	y ₃
Maximum	0.7	0.8	0.7	y ₂
Minimum	0.5	0.2	0.1	y ₁
Average	0.56	0.46	0.42	y ₁
Trim 20% Average	0.53	0.43	0.43	y ₁



Continuous-valued Output

◆ Example:

- 3 classes
- 5 base classifiers

$$D(x) = \begin{bmatrix} 0.6 & 0.4 & 0.1 \\ 0.7 & 0.2 & 0.7 \\ 0.5 & 0.2 & 0.1 \\ 0.5 & 0.7 & 0.6 \\ 0.5 & 0.8 & 0.6 \end{bmatrix}$$

Weight Average
L Weight

y₁

$$w = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.1 \\ 0.1 \\ 0.2 \end{bmatrix}$$

Class 1	0.4 × 0.6 + 0.2 × 0.7 + 0.1 × 0.5 + 0.1 × 0.5 + 0.2 × 0.5 = 0.12
Class 2	0.4 × 0.4 + 0.2 × 0.2 + 0.1 × 0.2 + 0.1 × 0.7 + 0.2 × 0.8 = 0.09
Class 3	0.4 × 0.1 + 0.2 × 0.7 + 0.1 × 0.1 + 0.1 × 0.6 + 0.2 × 0.6 = 0.07



Continuous-valued Output

◆ Example:

- 3 classes
- 5 base classifiers

Weight Average
L x c Weight

y_2

$$D(x) = \begin{bmatrix} 0.6 & 0.4 & 0.1 \\ 0.7 & 0.2 & 0.7 \\ 0.5 & 0.2 & 0.1 \\ 0.5 & 0.7 & 0.6 \\ 0.5 & 0.8 & 0.6 \end{bmatrix}$$

$$w = \begin{bmatrix} 0.1 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.4 \\ 0.2 & 0.1 & 0.1 \\ 0.4 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.2 \end{bmatrix}$$

Class 1 $0.1 \times 0.6 + 0.1 \times 0.7 + 0.2 \times 0.5 + 0.4 \times 0.5 + 0.2 \times 0.5 = 0.11$

Class 2 $0.2 \times 0.4 + 0.1 \times 0.2 + 0.1 \times 0.2 + 0.2 \times 0.7 + 0.4 \times 0.8 = 0.12$

Class 3 $0.2 \times 0.1 + 0.4 \times 0.7 + 0.1 \times 0.1 + 0.1 \times 0.6 + 0.2 \times 0.6 = 0.10$



Diversity

- If all base classifiers **always have the same decision**, no need to consider all of them
- Diversity** is a measure of **difference between base classifiers**
 - An intuitive, key concept for ensemble
 - Many definitions
- Can be categorized according to output type: **Label** and **Continuous-valued Output**



Label Output

◆ Pairwise Method

Consider two base classifiers D_i and D_k

◆ There are four different possibilities:

	D_k correct	D_k wrong
D_i correct	N^{11}	N^{10}
D_i wrong	N^{01}	N^{00}

$$N = N^{00} + N^{01} + N^{10} + N^{11}$$

N^{11} : Number of times when two base classifiers are correct
 N^{10} : Number of times when a classifier is correct and another is wrong
 N^{01} : Number of times when a classifier is wrong and another is correct
 N^{00} : Number of times when two base classifiers are wrong
 N : Total Number of times



Label Output

◆ Disagreement Measure

- Probability two classifiers **disagree each other**
- Range: **0 - 1** (most diverse, totally disagree)

$$\frac{N^{01} + N^{10}}{N}$$

◆ Double Fault Measure

- Probability two classifiers **being wrong together**
- Range: **0 - 1** (most diverse, both wrong all the time)

$$\frac{N^{00}}{N}$$



Continuous-valued Output

- ◆ Correlation Coefficient (CC)
 - CC between two classifiers' outputs (pairwise)
 - Diversity is the average of CCs of $L(L-1)/2$ pairs
 - 1: not diverse (identical)
 - 0: independent
 - -1: the most diverse
 - Definition: $CC(f_i, f_j) = \frac{E[(f_i - \mu_{f_i})(f_j - \mu_{f_j})]}{\sigma_{f_i} \sigma_{f_j}}$
 - f_i : the i th classifiers' outputs
 - μ_{f_i} and σ_{f_i} : mean and standard deviation of the i th classifiers' output on all samples



Diversity

- ◆ How to make base classifiers **diversify**?
 - Implicit Method
 - Using different Training Sets
 - Samples
 - Features
 - Using different Base Classifiers
 - Learning Models
 - Training Parameters
 - Explicit Method
 - Maximize diversity during training



Ensemble Construction Method

- ◆ The most well known MCS construction methods:
 - Bagging
 - Boosting
 - Negative Correlation



Bagging

- ◆ Aim to aggregated different versions of a model generated by **bootstrap** training samples
 - **Bootstrapping**: use samples of the data with repetition
- ◆ **Algorithm**
 - Random a **replicate of training set with replacement**
 - **Train** a base classifier using the **replicate**
 - Repeat until L number of base classifiers are trained
 - Finally, **voting or average** fusion (no weighting) method can be used



Example

◆ Bagged Decision Trees

- Draw bootstrap samples to form **sample sets**
- Train trees on each sample set
- Average prediction of trees on unseen samples

◆ Random Forests (Bagged Trees++)

- Draw bootstrap samples to form **sample sets**
- Each set uses different **feature subsets**
- Train trees on each samples
- Average prediction of trees on unseen samples



Bagging

◆ Advantage:

- Simple, **easy to understand**
- **Good for unstable classifier**
 - If small changes in the training set causes large difference in the generated classifier
 - The algorithm has high variance
 - E.g. Decision Tree

◆ Disadvantage:

- May **not** generating **complementary** base models



Boosting

◆ Actively **generate complementary base models**

- Train the next base model based on mistakes made by previous models

◆ Generate a sequence of base models each **focusing on previous one's errors**



Example

◆ **Adaptive Boosting (AdaBoost)**

- Base models are **trained by minimizing the weighted error**
 - A larger weight is assigned to samples classified wrongly
- **Weighted average** is used as the fusion method
 - A model with a smaller error is assigned a larger weight; vice versa





- ◆ For continuous-valued output base classifiers, e.g. MLPNN
- ◆ Explicitly consider diversity measure
- ◆ Objective Function per each base classifier:

$$\text{Objective Function} = \text{Training Error} + \lambda \text{ Diversity}$$

λ is the tradeoff parameter

$$\frac{1}{N} \sum_{i=1}^N \left(f_k(x_i) - f(x_i) \left[\sum_{\substack{j=1 \\ j \neq k}}^L (f_j(x_i) - f(x_i)) \right] \right)$$

where $f(x_i) = \frac{1}{L} \sum_{j=1}^L f_j(x_i)$