Multiple Classifier Systems

Prof. Daniel Yeung
School of Computer Science and Engineering
South China University of Technology

Outline

- Introduction (Ch3)
- Fusion (Ch4 and Ch5)
- Diversity (Ch10)
- MCS Construction Methods (Ch7)

Why MCS?

In practice, many classifiers with different settings are trained for a given classification problem (dataset)

Reference book for this Lecture

- Combining Pattern Classifiers
  - Ludmila I. Kuncheva
  - Wiley
Why MCS? An example

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

10% Training 90% Testing

Select for training Reserve for evaluation

Let’s try to use a very simple MLP Neural Network with one hidden layer

However, we still have no idea how many hidden neurons should be used

So, we have trained many 3-layer MLPNNs with different settings
Example

- MLPNN 11 has the best performance on unseen data
  - But we only know the training error

- How to choose the most suitable MLPNN?
  - Selection Criteria:
    - Training Accuracy
      - Many choices
    - Training Accuracy + Complexity
      - Smallest number of hidden neurons?

- Which criterion is the best?

Example

- Why not combining different MLPNNs?

Multiple MLPNNs – Multiple Classifier System

- Average of continuous outputs of 18 classifiers

<table>
<thead>
<tr>
<th>#</th>
<th>HN</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>0.020</td>
<td>0.051</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>0.010</td>
<td>0.040</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>0.010</td>
<td>0.021</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.010</td>
<td>0.029</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>0.010</td>
<td>0.041</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0.010</td>
<td>0.039</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>0.010</td>
<td>0.040</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0.010</td>
<td>0.024</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>0.010</td>
<td>0.021</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>0.140</td>
<td>0.161</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>0.010</td>
<td>0.018</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>0.160</td>
<td>0.152</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>0.010</td>
<td>0.018</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>0.010</td>
<td>0.019</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>0.010</td>
<td>0.029</td>
</tr>
<tr>
<td>16</td>
<td>12</td>
<td>0.010</td>
<td>0.023</td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>0.010</td>
<td>0.027</td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>0.010</td>
<td>0.021</td>
</tr>
</tbody>
</table>

Training Error = 0.0100
Testing Error = 0.0167

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

Why? Say 12 of 18 classifiers have outputs around 0.55 (which means class A, but the remaining 6 classifiers have outputs around 0.1, which means class B). So average output is around 0.4, which means class A as MCS final output

Example: Conclusion

- Major Drawbacks of selecting the BEST classifier
  1. Selecting 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
  2. Selecting a wrong classifier will definitely lead to erroneous classification result
  3. A single, trained classifier may not be adequate to handle today’s increasingly complex problems

- Combining classifiers may be a better choice

Multiple Classifier System (MCS)

- A MCS consists of a set of individual classifiers united by a fusion method which is used to combine their decisions to give a final output

Base classifier / Individual classifier

Fusion Method

Fusion

result

MCS

result
Multiple Classifier System (MCS)

1. Sample X is fed into each individual classifier
2. Each individual classifier makes its own decision
3. Final decision is made by combining all individual decisions

\[
\begin{align*}
X & \xrightarrow{f_1} \cdots \xrightarrow{f_L} \text{Fusion} \\
\text{Final Result} & 
\end{align*}
\]

MCS must be better than Single?
- All cases: **NO!**
- But in many cases it is true

Three factors affecting MCS performance

**Accuracy of base classifiers**
- The performance of a base classifier is affected by
  - Training Dataset (sample and feature)
  - Learning Model (Type of classifier)
  - Model’s Parameters (e.g. the number of neurons and layers in a NN)

**Diversity among individual classifiers**
- How different are the decisions reached by the classifiers?

Obviously, if the performances of base classifiers are poor, the MCS cannot be expected to have a good performance (hope it will be better than base classifiers)
Three factors affect MCS performance

Fusion Method

- A method to arrive at a group (MCS) decision
- Fusion method can be separated into two categories according to the individual classifier output types:
  - **Label output**
    - E.g. X is Class 1
    - Output: the class which the sample belongs to
  - **Continuous-valued output**
    - E.g. X is Class 1 0.7
      Class 2 0.3
    - Output a value (probability) for each class

Fusion method can be separated into two categories according to the individual classifier output types:

- **Label output**
  - E.g. X is Class 1
  - Output: the class which the sample belongs to
- **Continuous-valued output**
  - E.g. X is Class 1 0.7
    Class 2 0.3
  - Output a value (probability) for each class

Three factors affect MCS performance: Fusion Method

Label Output

- Base classifier $D_i$ produces a class label $D_i(x) = [d_{i,1}, d_{i,2}, \ldots, d_{i,c}]^T$
  
  where
  - $c$ is number of classes
  - $i$th column refers to $i$th classifier $d_{i,j}$ which is equal to 1 if $x$ belongs to class $j$, otherwise it is 0

- Example: 3-class problem, a MCS with 4 base classifiers and their decisions are class 2, class 3, class 1 and class 2 respectively, its output is

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

  - Classifier 2 decision
  - Decision on class 2 since 2 Out of 4 base classifiers select 2

Three factors affect MCS performance: Fusion Method

Label Output -- Decision Rule

- **Democracy Method**
  - **Unanimity** (all agree)
    - Many unknown cases
  - **Simple Majority** (50% + 1)
    - Many unknown cases

- **Majority Vote** (most votes)
  - Also called the Plurality
  - The class is $\omega_k$ if
    
    $$\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{L} \sum_{i=1}^{L} d_{i,j}$$

- Majority vote method assumes each base classifier has same classification ability
- However, in most cases, this is not true

- **Weighted Majority Vote**
  - Assign a weight ($w$) to each base classifier based on its ability (accuracy)
  - The class is $\omega_k$ if
    
    $$\sum_{i=1}^{L} w_i d_{i,k} = \max_{j=1}^{L} \sum_{i=1}^{L} w_i d_{i,j}$$
Three factors affect MCS performance: Fusion Method
Label Output -- Decision Rule

Example: 3 classes, 5 base classifiers

\[ D(x_1) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \]

\[ w = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.2 \\ 0.4 \\ 0.2 \end{bmatrix} \]

\( \omega_1 : 3 \)
\( \omega_2 : 1 \)
\( \omega_3 : 1 \)

Unanimity
Simple Majority
Majority Vote
Weighted Majority Vote

\[ D(x_2) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \]

\[ w = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.3 \\ 0.2 \end{bmatrix} \]

\( \omega_1 : 2 \)
\( \omega_2 : 1 \)
\( \omega_3 : 2 \)

Unanimity
Simple Majority
Majority Vote
Weighted Majority Vote

The output of \( i \)th individual classifier \( s_i \) can also be defined as \( [\omega_{l,k}]^T \)
where \( \omega_{l,k} \) is one of the classes, \( c \geq k \geq 1 \) and \( c \) is the number of classes

Example: 3-class problem, a MCS with 4 base classifiers and their decisions are class 2, class 3, class 1 and class 2 respectively, its output is

\[ s(x) = [\omega_2 \ \omega_3 \ \omega_1 \ \omega_2]^T \]

Behavior Knowledge Space Method (BKS)

A posterior probability \( P(\omega_k | s) \) look-up table (BKS table) is created from the training set

\( s(x) \) is class \( \omega_k \) if

\[ P(\omega_k | s) = \max_{j=1}^{c} P(\omega_j | s) \]

2 classes
3 base classifier
150 training samples

When classifier 1 is \( \omega_1 \), classifier 2 is \( \omega_2 \), classifier 3 is \( \omega_3 \)

4 training samples are \( \omega_1 \) with \( P(\omega_1 | s) = 4/24 \)
20 training samples are \( \omega_2 \) with \( P(\omega_2 | s) = 20/24 \)

For an unseen sample \( x \), if \( s(x) = (\omega_2, \ \omega_2, \ \omega_3) \)
The decision of MCS is \( \omega_2 \)

\( s_1 \) \( s_2 \) \( s_3 \) \( \omega_1 \) \( \omega_2 \)
\( \omega_1 \ \omega_1 \ \omega_1 \ \omega_1 \ \omega_2 \)
\( \omega_1 \ \omega_1 \ \omega_2 \ \omega_2 \ \omega_1 \)
\( \omega_2 \ \omega_2 \ \omega_1 \ \omega_2 \ \omega_2 \)
\( \omega_2 \ \omega_2 \ \omega_1 \ \omega_2 \ \omega_2 \)
\( \omega_2 \ \omega_2 \ \omega_2 \ \omega_2 \ \omega_2 \)

\( \omega_2 \ \omega_2 \ \omega_2 \ \omega_2 \ \omega_2 \)
Three factors affect MCS performance: Fusion Method

Label Output -- Decision Rule

- **Behavior Knowledge Space Method (BKS), Example**
  - 2 classes
  - 3 base classifiers
  - 150 training samples

<table>
<thead>
<tr>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$\omega_1$</th>
<th>$\omega_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega_1$</td>
<td>$\omega_1$</td>
<td>$\omega_1$</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>$\omega_1$</td>
<td>$\omega_1$</td>
<td>$\omega_2$</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

When classifier 1 is $\omega_1$, classifier 2 is $\omega_2$, classifier 3 is $\omega_1$.

3 training samples are $\omega_1$:

- $P(\omega_1 | s) = 3/19$
- $\omega_2 \omega_1 \omega_1$ : 20 5
- $\omega_2 \omega_1 \omega_2$ : 2 19

16 training samples are $\omega_2$:

- $P(\omega_2 | s) = 16/19$
- $\omega_2 \omega_1 \omega_2$ : 3 22
- $\omega_2 \omega_2 \omega_1$ : 4 20

For an unseen sample $x$, if $s(x) = (\omega_1, \omega_2, \omega_1)$, the decision of MCS is $\omega_2$.

This decision ignores what majority suggests.

Continuous-valued output -- Decision Rule

- **Statistical Operator (for class $\omega_j$)**
  - **Product**
    \[ \mu_j(x) = \prod_{i=1}^{L} d_{i,j}(x) \]
  - **Median**
    \[ \mu_j(x) = \text{median}\{d_{i,j}(x)\} \]
  - **Minimum**
    \[ \mu_j(x) = \min_i \{d_{i,j}(x)\} \]
  - **Maximum**
    \[ \mu_j(x) = \max_i \{d_{i,j}(x)\} \]
  - **Simple Mean**
    \[ \mu_j(x) = \frac{1}{L} \sum_{i=1}^{L} d_{i,j}(x) \]
  - **Trimmed Mean**
    - Values are sorted and K percent of the values are dropped on each side
    - Find the mean of remaining values

Weighted Average

- **$L$ weights**
  \[ \mu_j(x) = \sum_{i=1}^{L} w_i d_{i,j}(x) \]

- **c x L weights**
  - Weights are specific for each class per base classifier
    \[ \mu_j(x) = \sum_{i=1}^{L} w_{ij} d_{i,j}(x) \]
Three factors affect MCS performance: Fusion Method
Continuous-valued output -- Decision Rule

- Example:
  - 3 classes
  - 5 base classifiers

<table>
<thead>
<tr>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>Product</th>
<th>Median</th>
<th>Max</th>
<th>Min</th>
<th>Avg</th>
<th>T.Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.7</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.7</td>
<td>0.5</td>
<td>0.56</td>
<td>0.53</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2</td>
<td>0.2</td>
<td>0.7</td>
<td>0.8</td>
<td>0.01</td>
<td>0.4</td>
<td>0.8</td>
<td>0.2</td>
<td>0.46</td>
<td>0.43</td>
</tr>
<tr>
<td>0.1</td>
<td>0.7</td>
<td>0.1</td>
<td>0.6</td>
<td>0.6</td>
<td>0.00</td>
<td>0.6</td>
<td>0.7</td>
<td>0.1</td>
<td>0.42</td>
<td>0.43</td>
</tr>
</tbody>
</table>

- Decision Profiles (DP)
  - For an unseen \( x \), the decision is made by

\[
\mu_j(x) = S(D(x), DT_j)
\]

where \( S \) is the similarity function

\[
DT_j = \frac{1}{N} \sum_{z_k \in \omega_j} DP(z_k)
\]
Three factors affect MCS performance: Fusion Method
Continuous-valued output -- Decision Rule

**Decision Templates (DT)**

- Example
  - 2 classes
  - 3 base classifiers

- Unseen sample \( x \)

- Result

<table>
<thead>
<tr>
<th>( S_{ED} )</th>
<th>( S_{SD} )</th>
<th>( \mu_i(x) )</th>
<th>( \mu_l(x) )</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.96</td>
<td>0.50</td>
<td>0.93</td>
<td>0.53</td>
<td>( \omega_1 )</td>
</tr>
<tr>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
<td>( \omega_2 )</td>
</tr>
</tbody>
</table>

See computation next page

---

Three factors affect MCS performance
Continuous-valued output -- Decision Rule

**How to Compute** \( S_{ED}(DP(x), DT_j) \)?

C = 2 classes and L = 3 classifiers

\[
[DT_j - DP(x)]^2 = [(0.6-0.3)^2 + (0.4-0.7)^2 + (0.8-0.6)^2 + (0.2-0.4)^2 + (0.5-0.5)^2 + (0.5-0.5)^2]
\]

So

\[
1 - \frac{1}{(c \times L)} [DT_j - DP(x)]^2 = 1 - \frac{0.18 + 0.08}{6} = 0.96
\]

---

Three factors affect MCS performance
Continuous-valued output -- Decision Rule

**Diversity**

- If all base classifiers in a MCS give the same decision, MCS adds no decision value
- Difference between base classifiers is necessary

**Diversity** is a measure of this difference

- An intuitive, key concept for MCS
- Many definitions
Three factors affect MCS performance

Diversity

- Diversity can be categorized as:
  - Label output
    - Pairwise
    - Non-Pairwise
  - Continues-valued output

Label Output: Pairwise Measures

- For two base classifiers $D_i$ and $D_k$
- There are four different possibilities:

<table>
<thead>
<tr>
<th>$D_i$ correct</th>
<th>$D_k$ correct</th>
<th>$D_i$ wrong</th>
<th>$D_k$ wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N^{11}$</td>
<td>$N^{10}$</td>
<td>$N^{01}$</td>
<td>$N^{00}$</td>
</tr>
</tbody>
</table>

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

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

Disagreement Measure

- Measure the probability that two classifiers will disagree on each other
- Range: 0 – 1
- 1 most sensitive when $N^{01} + N^{10} = N$
  - i.e. two classifiers totally disagree with each other

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

Double Fault Measure

- Measure the probability both classifier being wrong together
- Range: 0 – 1
- 1 most sensitive (when both are wrong all the time)

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

If a MCS has L base classifiers, there will be $L(L-1)/2$ pairs

The diversity measure of the MCS is the mean of pairwise diversity values of those pairs
Three factors affect MCS performance: Diversity
Label Output: Non-Pairwise Measures

**Entropy Measure**
(Entropy is a measure of the uncertainty associated with a random variable)
- If all base classifiers are the same, no diversity
- If all base classifiers are different, very diverse
- This concept is similar to Entropy

\[
E = \frac{1}{N} \sum_{k=1}^{N} \left\{ \min_{j=1}^{L} \left\{ \sum_{i=1}^{L} y_{i,j}(x_k), L - \sum_{i=1}^{L} y_{i,j}(x_k) \right\} \right\}
\]

\[
y_{i,j}(x_k) = \begin{cases} 
1, & \text{if the output of } i\text{th base classifier for sample } k \text{ is class } j \\
0, & \text{otherwise}
\end{cases}
\]

Three factors affect MCS performance: Diversity
Continued-valued output

**Correlation coefficient**
- the correlation coefficients (CC) between classifier outputs
  - One CC for each class between each pair of classifiers. Diversity is the average of these coefficients.

Definition

\[
\rho_{X,Y} = \text{corr}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}
\]

where X and Y are the two base classifiers’ outputs

**Measure of “difficulty”**
- Define \( \lambda \) taking values in \( \{0/L, 1/L, \ldots, L/L\} \)
  - \( \lambda \) denoting proportion of classifiers in \( D \) that correctly classify the training set
  - Diversity is measured by \( \text{var}(\lambda) \)
- Higher value, worse MCS (so right most plot shows best MCS)
  - If some points are difficult for all classifiers, and other points are easy for all classifiers, then NO Diversity (middle plot)
  - If points that are difficult for some classifiers but easy for other classifiers, Better Diversity (right plot)
  - If each point is equally difficult for all classifiers, there is a Diversity, but not likely a good one (left plot)

L=7 & N=100 points & \( p = 0.6 \) accuracy for all 7 classifiers

Best Diversity

Three factors affect MCS performance: Diversity
Diversity

**How to make base classifiers **diversify**?**

- Using different Training Sets
  - Samples
  - Features
- Using different Base Classifiers
  - Learning Models
  - Training Parameters
MCS Construction Method

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

MCS Construction Method

Bagging

Use bootstrapping to generate \( L \) training sets
- Draw \( L \) training sets at random with replacement
- Train one base classifier with each training set
- Voting is used as fusion

Algorithm
- Random a portion of full training set with replacement
- Train a base classifier using that portion
- Repeat until \( L \) number of base classifiers have been trained
- Finally, voting is used to combine these \( L \) base classifiers

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, MLPNN

Disadvantage:
- Generating complementary base classifiers is left to chance

Boosting

Actively generate complementary base classifiers

Train the next base classifier based on mistakes made by previous classifiers

Example: Adaptive Boosting (AdaBoost):
- Generate a sequence of base classifiers each focusing on previous one’s errors
- Base classifier is trained by minimizing the weighted error
  - A larger weight is assigned to samples classified wrongly
- Weighted average is used as fusion method
MCS Construction Method

Boosting: AdaBoost

Given: \((x_1, y_1), \ldots, (x_m, y_m)\) where \(x_i \in X, y_i \in Y = \{-1, +1\}\)

- **Initialisation**: \(D_1(i) = \frac{1}{m}\)  
  * Initial by equal weight

For \(t = 1, \ldots, T\):

1. Find the classifier \(h_t : X \rightarrow \{-1, +1\}\) that minimizes the error with respect to the distribution \(D_t\):
   \[
   h_t = \arg \min_{h \in H} \sum_{i=1}^{m} D_t(i) \mathbb{I}[y_i \neq h(x_i)]
   \]
   * Minimize the weighted error

2. Prerequisite: \(\varepsilon_t < 0.5\), otherwise stop

3. Choose \(\alpha_t \in \mathbb{R}\), typically \(\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}\) where \(\varepsilon_t\) is the weighted error rate of classifier \(h_t\)

4. Update:
   \[
   D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t h_t(x_i))}{Z_t}
   \]
   * Adjust the weight according to the error of base classifier
   where \(Z_t\) is a normalisation factor (chosen so that \(D_{t+1}\) will be a distribution)

Output the final classifier:

\[
H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)
\]

Weight in Weighted fusion

MCS Construction Method

Negative Correlation

- For continuous-valued output base classifiers, e.g. MLPNN
- Explicitly consider diversity measure

Objective Function per each base classifier:

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

\[
\frac{1}{N} \sum_{i=1}^{N} (f_i(x_i) - F(x_i))^2
\]

\[
\frac{1}{N} \sum_{i=1}^{N} \left( f_i(x_i) - f_{mcs}(x_i) \right) \left( \sum_{j \neq k} \left( f_j(x_i) - f_{mcs}(x_i) \right) \right)
\]

\(\lambda\) is the tradeoff parameter

\[
f_{mcs}(x_i) = \frac{1}{L} \sum_{j=1}^{L} f_j(x_i)
\]