Hierarchical Multiple Classifier Learning System

Yu-Yu Chou

August 31, 1999
Motivation

- Automatic learning is necessary for many applications to reduce the development costs.
- Current machine learning algorithms do not scale well for complicated data or large amounts of data.
- New algorithms need to be investigated to handle the increasing amount and complexity of data.
Problem Description

- **Application:** automatic prescreening for cervical cancer examination - NeoPath Inc.

- **Current approaches:** multiple-level probabilistic decision trees created with extensive interaction and assistance from experts.

- **Goals:** by engaging various machine learning techniques to
  - Accelerate the training process.
  
  - Automate the training procedure and reduce human interaction.

  - Enhance the classification accuracy.
Problem Characteristics

- The amounts of data are tremendous.
- Each data instance (cell) is described by a set of sophisticated features.
- Multiple level classes outputs:
  - Level I classes: 3.
  - Level II classes: 16 (7).
  - Level II classes: 142.
- There are many different sources of noise in the data set.
  - technicians’ operating differences.
  - focus problems.
  - variations in specimen collection.
  - data collection procedures.
Related Literature

Stand-alone classification algorithms

• Decision Trees: C4.5 - Quinlan (1993).
• Rule-Based Induction: CN2 - Clark (1989).
• Instance-Based Learning.
• Neural Networks: NevProp (1998).
Related Literature (contd.)

Construction of Ensembles of Classifiers

- Subsampling the Training Data: Bagging - Breiman (1996); Boosting - Schapire (1995)
- Manipulating the Target Function.
- Injecting Randomness.

Methods for Combining Classifiers

- Unweighted or Weighted Vote.
Our Philosophy

- Multiple Classifier System.

- Constructing Ensembles of Classifiers:
  - Manipulating the training data distribution: Data clustering.
  - Manipulating the target function: Subclass labeling.

- Combining Classifiers: cross-validation super-classifiers.
System Diagram

TRAINING PHASE

TRAINING DATA

DATA CLUSTERING

SUBCLASS LABELING

ENSEMBLE CONSTRUCTION

CROSS-VALIDATION

SUPERCLASSIFIER CONSTRUCTION

ERROR INSTANCES

FEATURE SELECTION

CLASSIFICATION PHASE

TEST DATA

ENSEMBLE CLASSIFIERS

SUPERCLASSIFIER

RESULTS
Data Clustering

To change the distribution of training data and reduce the training cost of the component classifiers.

- Random Partitioning.
- Graph-Theoretic Clustering: Shapiro & Haralick (1979).
Graph-Theoretic Clustering

simple polygonal shape

corresponding relational graph
Subclass Labeling Concept

To improve the estimation of decision boundaries.

The original data points of a 2-class example
Subclass Labeling Concept (contd.)

classified by neural net without sub-classes
Subclass Labeling Concept (contd.)

classified by neural net with sub-classes
Component Classifier Construction

Level II Classes

TRAINING DATA

CLUSTERING ALGORITHM

CLUSTER INFO

DATA CLUSTERS

COMPONENT CLASSIFIERS
Super-classifier Construction

Level II Classes

TRAINING DATA (TEST DATA) → CLUSTER INFO → RANK → ERROR INSTANCES

C1 → C2 → ... → CN

Level I Classes

NEW DATA → SUPER-CLASSIFIER

Optional / Alternative
Error Instances Detection

Training Stage

- Training Data
  - Super Classifier 1
    - Correct Instances
    - Error Instances
      - Error Instance Detection Classifier
        - Training Group A
        - Training Group B

Test Stage

- Test Data
  - Error Instance Detection Classifier
    - Test Group A
    - Test Group B
      - Super Classifier 1
      - Super Classifier 2

Assign New Classes
Experiment Settings

- Data Sets:
  - NeoPath-1: 19,125 cases (323).
  - NeoPath-2: 24,345 cases (291).
  - Features are all continuous values.

- Training Set: 60% of cases; Test Set: 40%.

- Base-line Classification Algorithms:
  - Decision Tree Classifier: C4.5.
  - Backpropagation Neural Networks: NevProp.

- Clustering Algorithms:
  - Random Partitioning.
  - K-means Clustering.
  - Graph-Theoretic Clustering.
Experiment Settings (contd.)

- Output Classes:

<table>
<thead>
<tr>
<th></th>
<th>First</th>
<th>Second</th>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abnormal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ascus</td>
<td></td>
<td>2625</td>
<td>5024</td>
<td></td>
</tr>
<tr>
<td>LSIL</td>
<td></td>
<td>2732</td>
<td>3443</td>
<td></td>
</tr>
<tr>
<td>HSIL</td>
<td></td>
<td>3968</td>
<td>3229</td>
<td></td>
</tr>
<tr>
<td>Cancer</td>
<td></td>
<td>1533</td>
<td>3516</td>
<td></td>
</tr>
<tr>
<td>Repair</td>
<td></td>
<td>1477</td>
<td>2404</td>
<td></td>
</tr>
<tr>
<td>Normal</td>
<td></td>
<td>5040</td>
<td>3775</td>
<td></td>
</tr>
<tr>
<td>Artifact</td>
<td></td>
<td>1750</td>
<td>2954</td>
<td></td>
</tr>
</tbody>
</table>

- Result Definition:
  - **Sensitivity** - the percentage of abnormal cases classified as abnormal.
  - **Specificity** - the percentage of normal cases classified as normal.
System Evaluation (NeoPath-1)

Sensitivity vs Specificity Plot (Algorithm: NevProp)

Performance of Super-classifier Alone

Various settings for target class and number of classifier.
System Evaluation (NeoPath-1)

Sensitivity vs Specificity Plot (Algorithm: NevProp)

Different Clusters Settings

5 Clusters
10 Clusters
14 Clusters
20 Clusters

Various settings for different clusters.
System Evaluation (NeoPath-1)

Sensitivity vs Specificity Plot (Algorithm: NevProp)

Different Feature Subsets

Sensitivity (Percentage of Abnormals classified as Abnormal)

Specificity (Percentage of Normals classified as Normal)

Various settings for different feature sets.
System Evaluation (NeoPath-2)

Sensitivity vs Specificity Plot (Algorithm: NevProp)

Various settings for target class and number of classifier.
System Evaluation (NeoPath-2)

Sensitivity vs Specificity Plot (Algorithm: NevProp)

Different Clusters Settings

Various settings for different clusters.
System Evaluation (NeoPath-2)

Sensitivity vs Specificity Plot (Algorithm: NevProp)

Various settings for different feature sets.
System Evaluation (Forest Cover Data)

- Source: UCI Knowledge Discovery in Databases Archive.
- Data Description: 11,340 (training) + 3780 (validation) + 565,892 (test) = 581,012 cases with 54 features and 7 output classes.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Discriminant Analysis</td>
<td>58</td>
</tr>
<tr>
<td>Backpropagation</td>
<td>70</td>
</tr>
<tr>
<td>NevProp</td>
<td>23.96</td>
</tr>
<tr>
<td>C4.5</td>
<td>63.64</td>
</tr>
<tr>
<td>NeuNet Pro SFAM</td>
<td>68</td>
</tr>
<tr>
<td>Hierarchical Multiple Classifier</td>
<td>70.81</td>
</tr>
</tbody>
</table>

*a ≈ twice the number of training records than the other experiments."
Comparison of Different Clustering Algorithms

Classification accuracy of the NeoPath-2 test data with a full set of 291 features.

\(^a\) The priority of identifying the abnormal cases is much higher than the normal cases.
Comparison of Different Clustering Algorithms

Classification accuracy of the NeoPath-2 test data with a subset of 74 features.
Comparison of Different Classification Algorithms

Classification accuracy of the NeoPath-2 test data with a subset of 74 features.
Comparison of Different Classification Algorithms

Sensitivity vs Specificity Plot

Sensitivity-Specificity plot for various classifier algorithms.
Contributions

- Described a flexible hierarchical multiple classifier system to meet the needs of different applications.

- Provided an efficient, low cost and high accuracy solution for complicated classification problems through data clustering and subclass labeling.

- Minor Contribution: Utilized the component classifiers as a type of feature selector.
Future Work

- Investigate various algorithms for combining the results of component classifiers.

- Investigate the erroneous instance detection procedure to better identify the instances with low probabilities to be correctly classified.

- Adaptation of other classification and clustering algorithms for different applications.