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.
- Hybrid System: RISE - Domingos (1995).
- Neural Networks: NevProp (1998).

Related Literature (contd.)

Construction of Ensembles of Classifiers

- Subsampling the Training Data: Bagging - Breiman (1996); Boosting - Schapire (1995)
- Manipulating the Input Features: Random Subspace Decision Forests - Ho (1998).
- Manipulating the Target Function.
- Injecting Randomness.

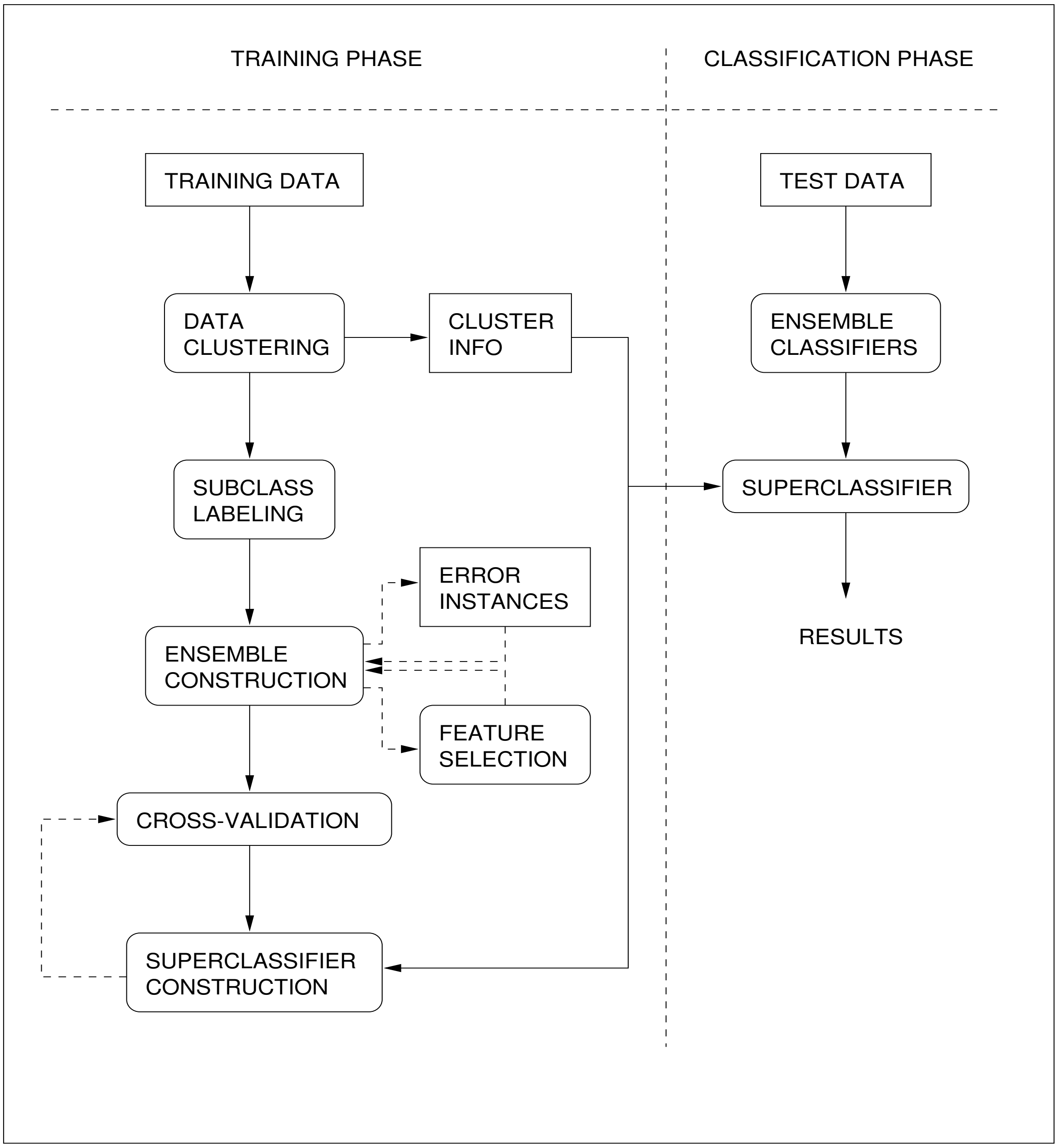
Methods for Combining Classifiers

- Unweighted or Weighted Vote.
- Gating Network: EM Algorithm - Jordan & Jacob (1994).
- Stacking - Breiman (1996).

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

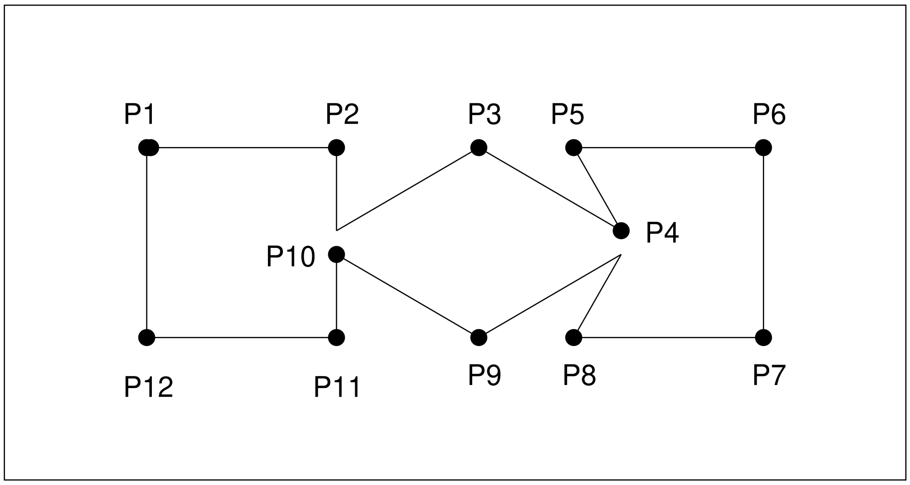


Data Clustering

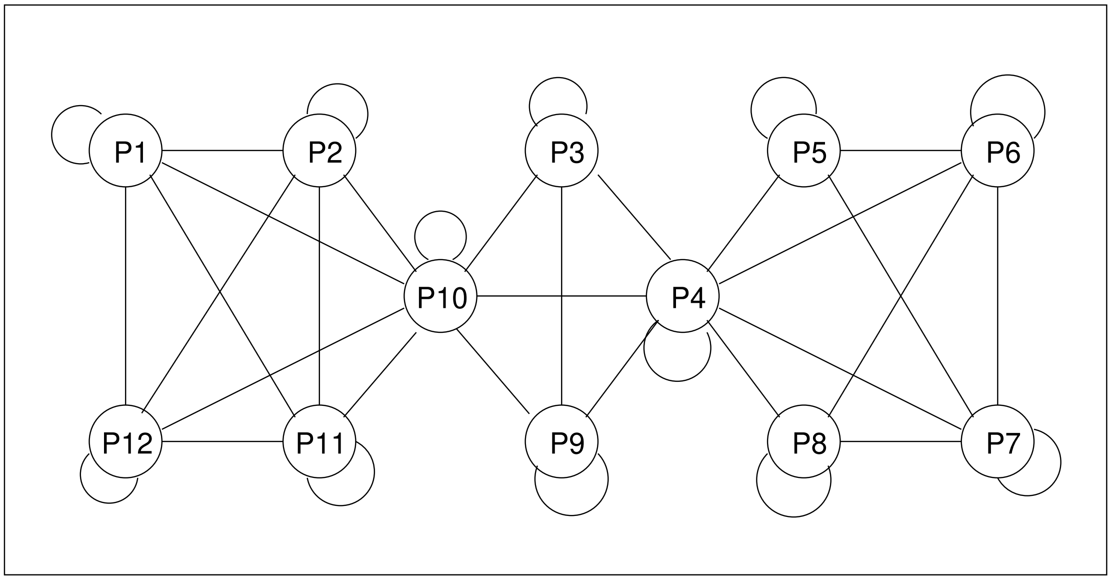
To change the distribution of training data and reduce the training cost of the component classifiers.

- Random Partitioning.
- K-means Clustering: Heng (1996).
- Graph-Theoretic Clustering: Shapiro & Haralick (1979).

Graph-Theoretic Clustering



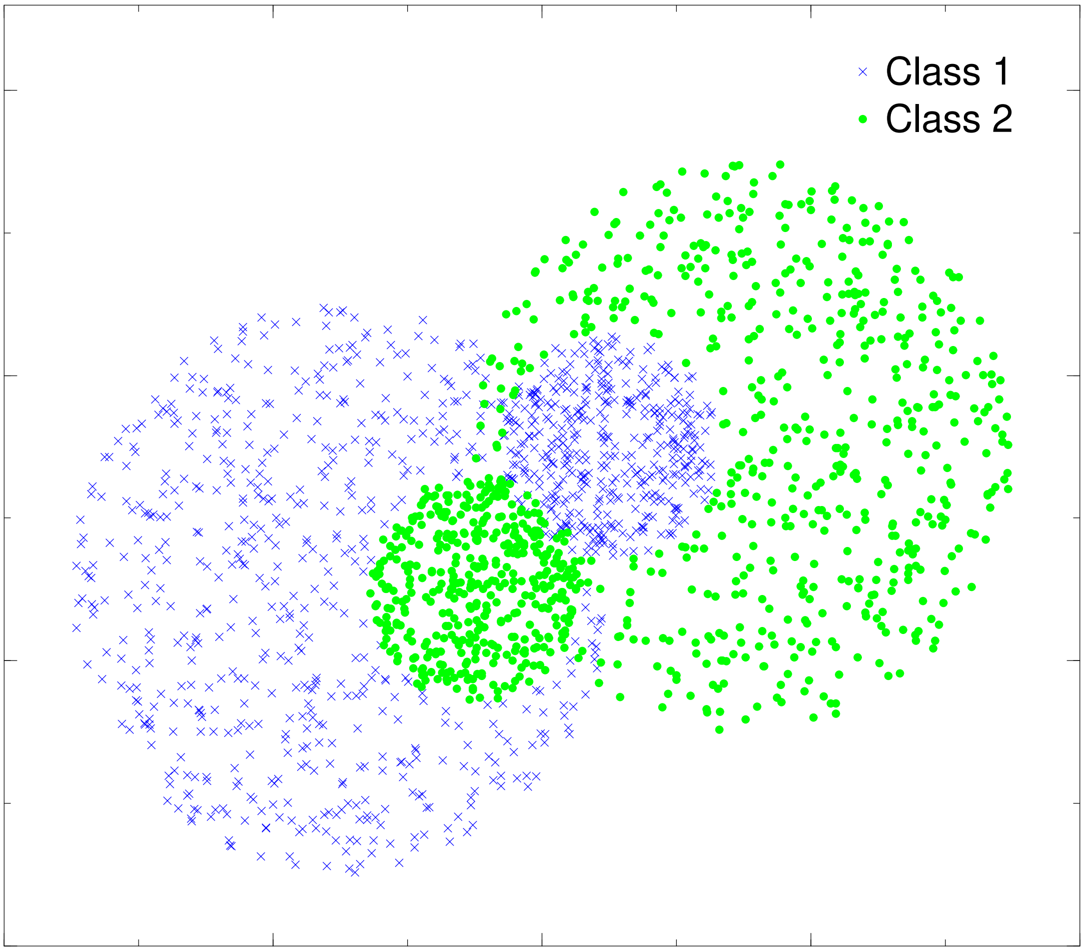
simple polygonal shape



corresponding relational graph

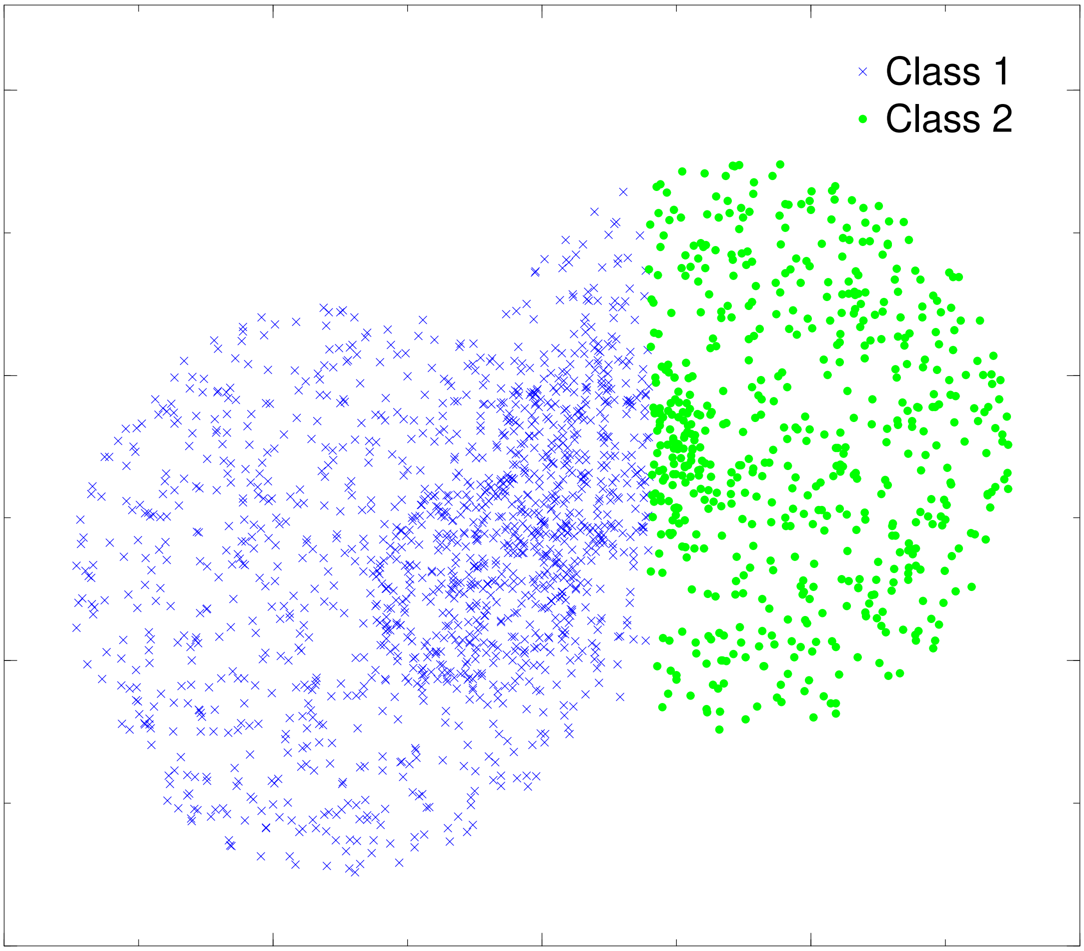
Subclass Labeling Concept

To improve the estimation of decision boundaries.



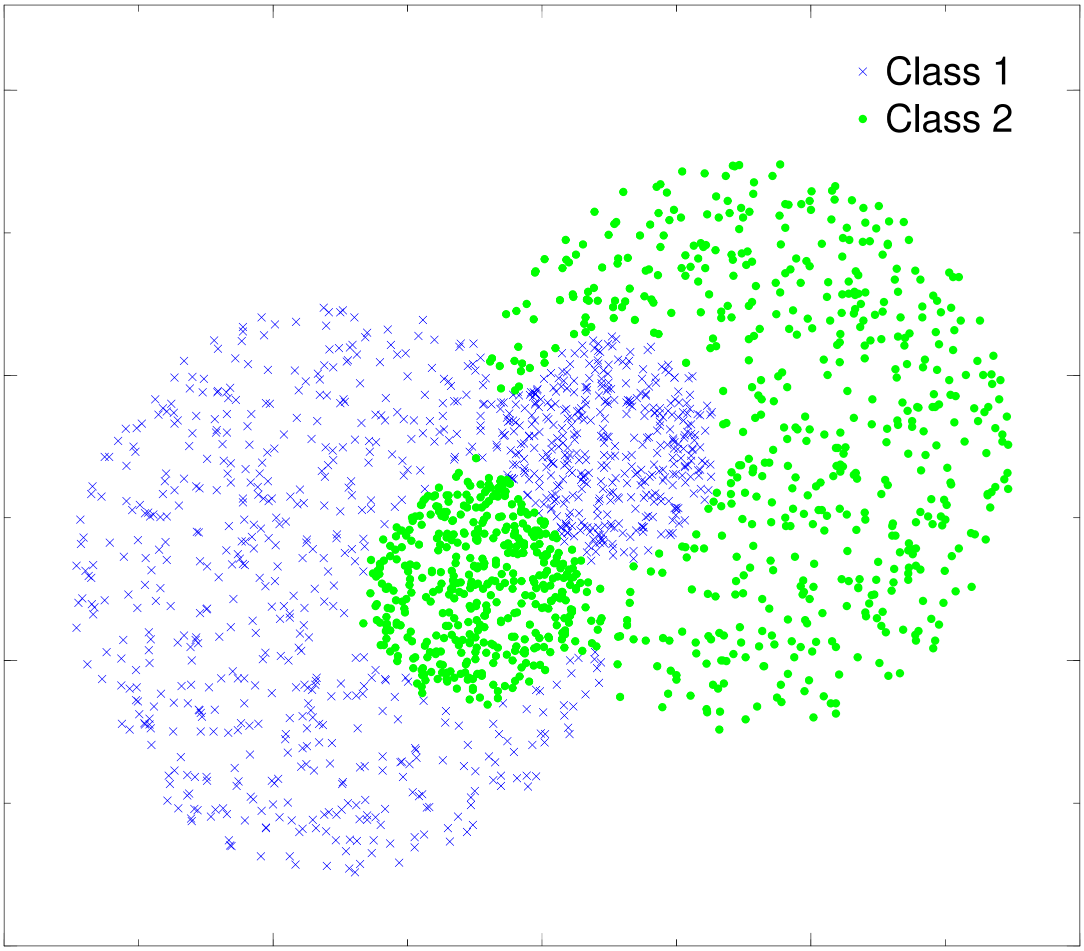
The original data points of a 2-class example

Subclass Labeling Concept (contd.)



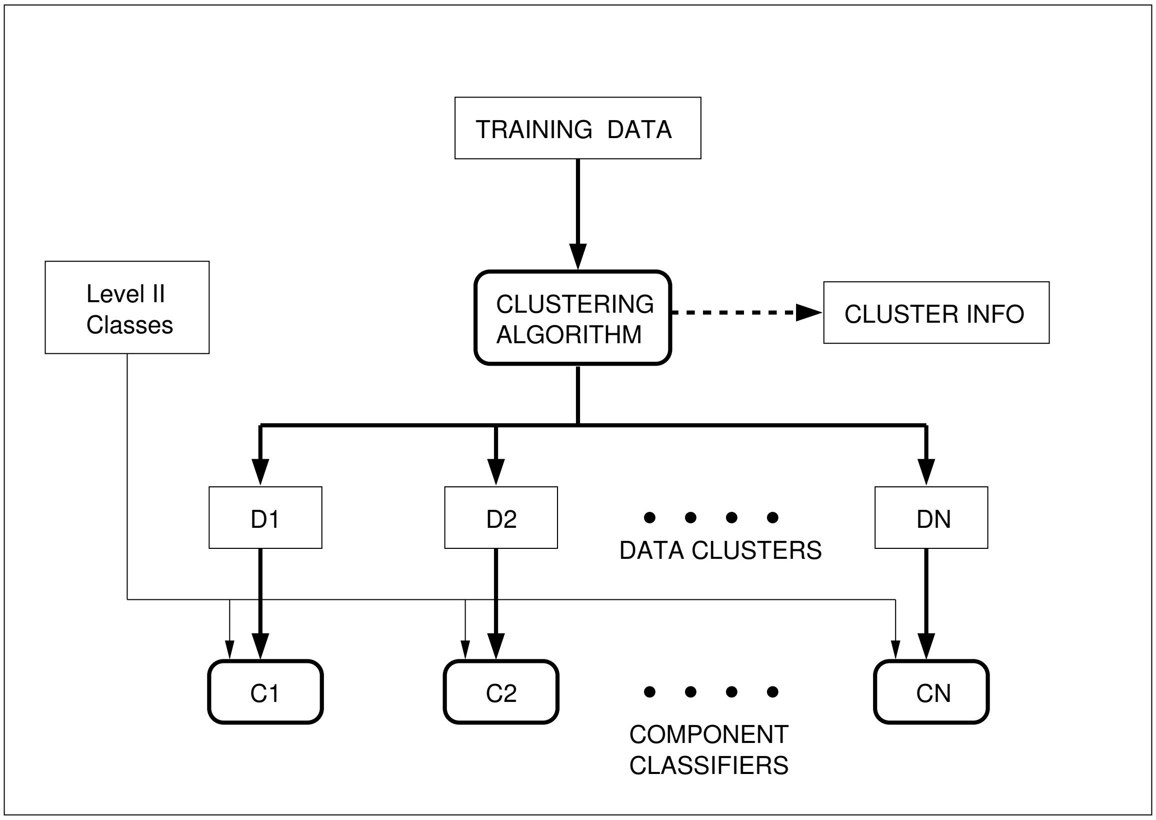
classified by neural net without sub-classes

Subclass Labeling Concept (contd.)

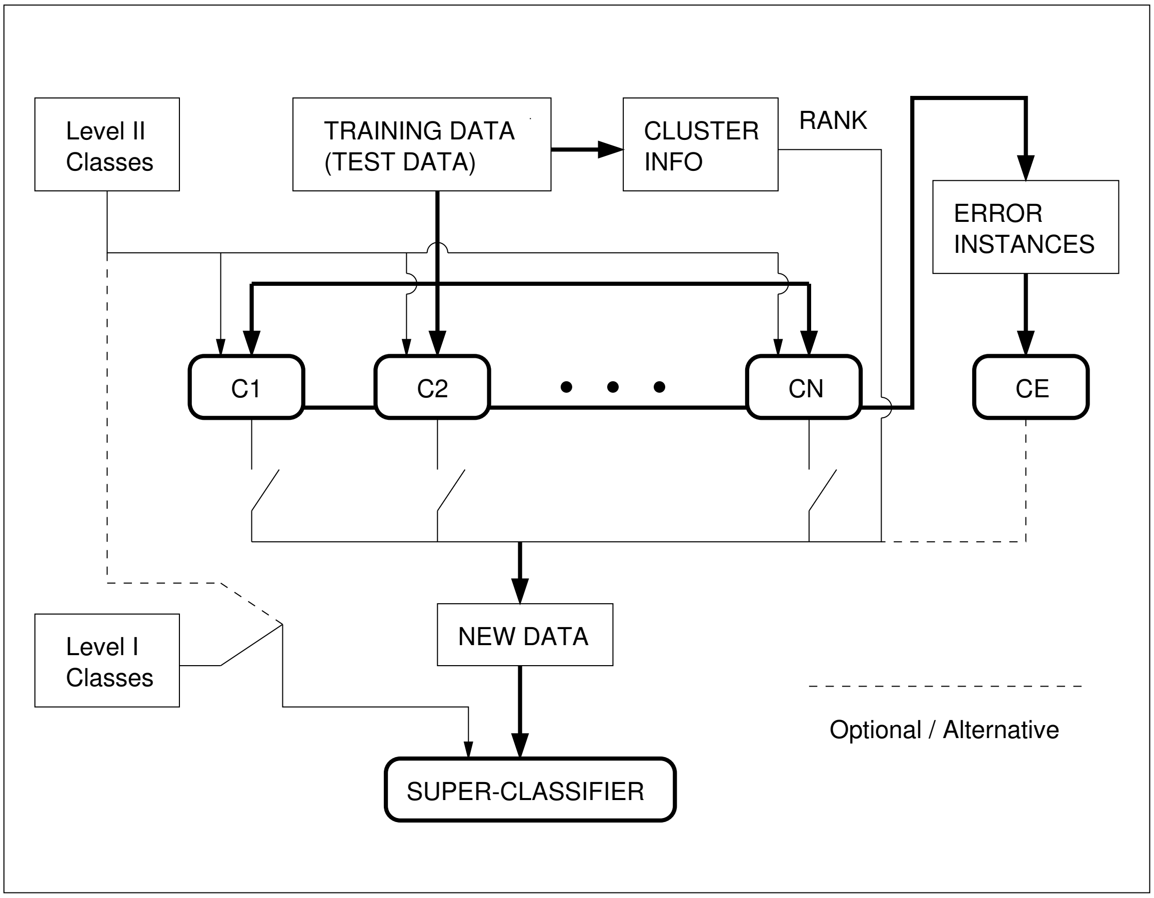


classified by neural net with sub-classes

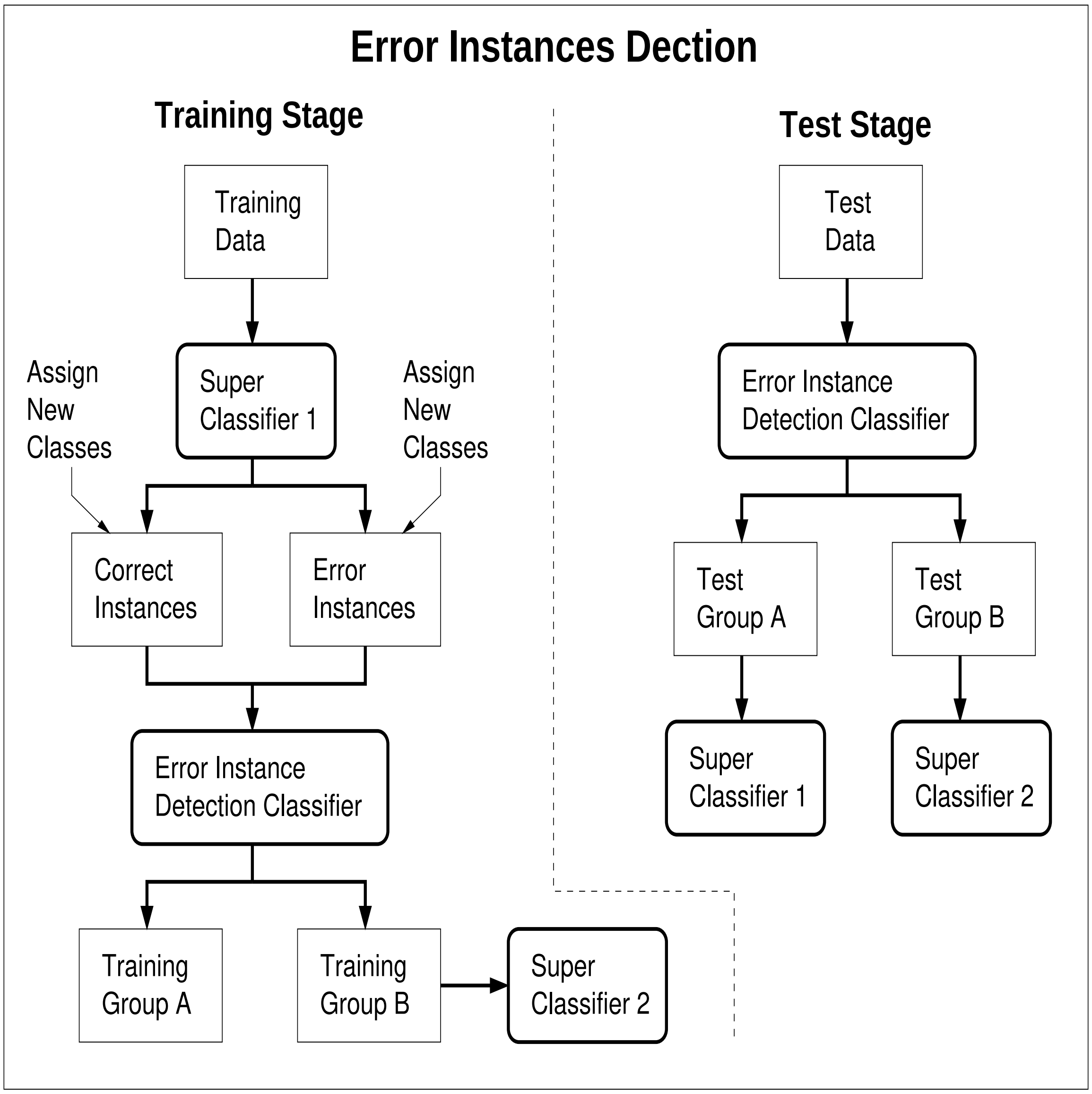
Component Classifier Construction



Super-classifier Construction



Error Instances Detection



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.)

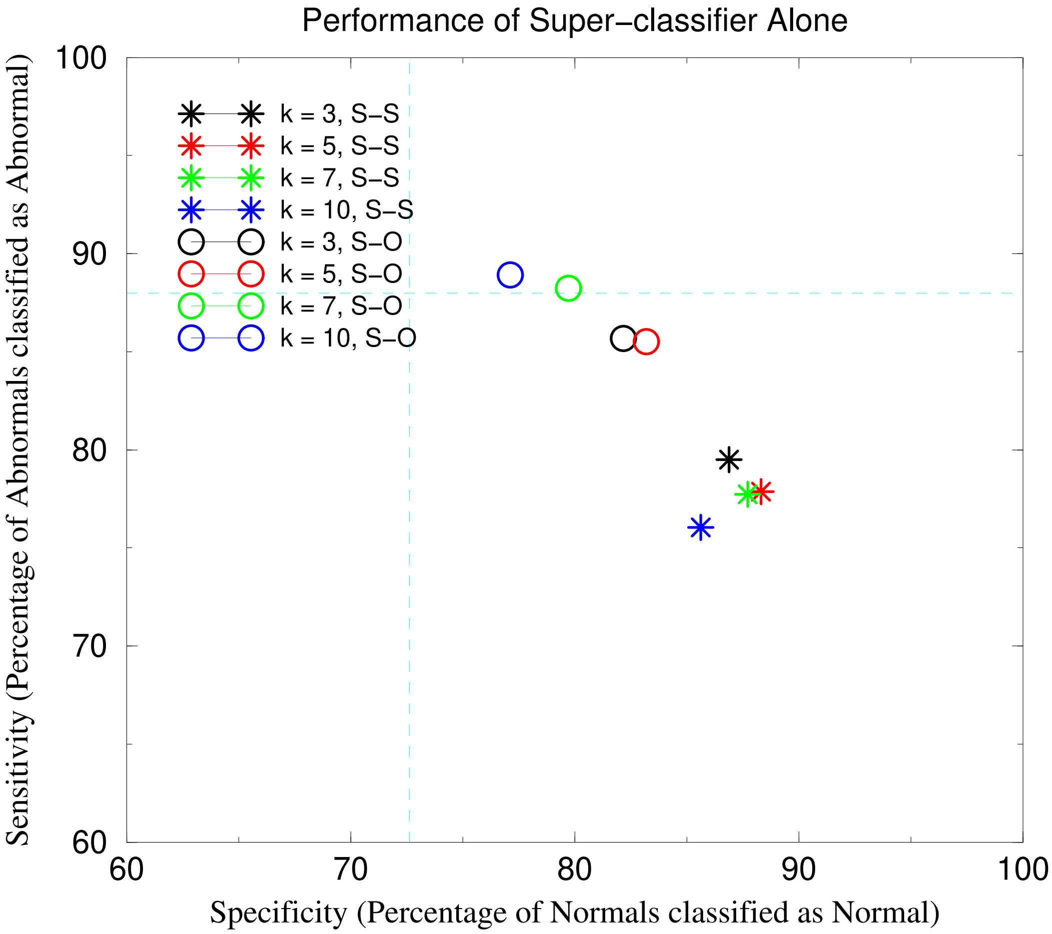
- Output Classes:

First	Second	Set 1	Set 2
Abnormal	Ascus	2625	5024
	LSIL	2732	3443
	HSIL	3968	3229
	Cancer	1533	3516
	Repair	1477	2404
Normal		5040	3775
Artifact		1750	2954

- Result Definition:
  - **Sensitivity** - the percentage of abnormal cases classified as abnormal.
  - **Specificity** - the percentage of normal cases classified as normal.

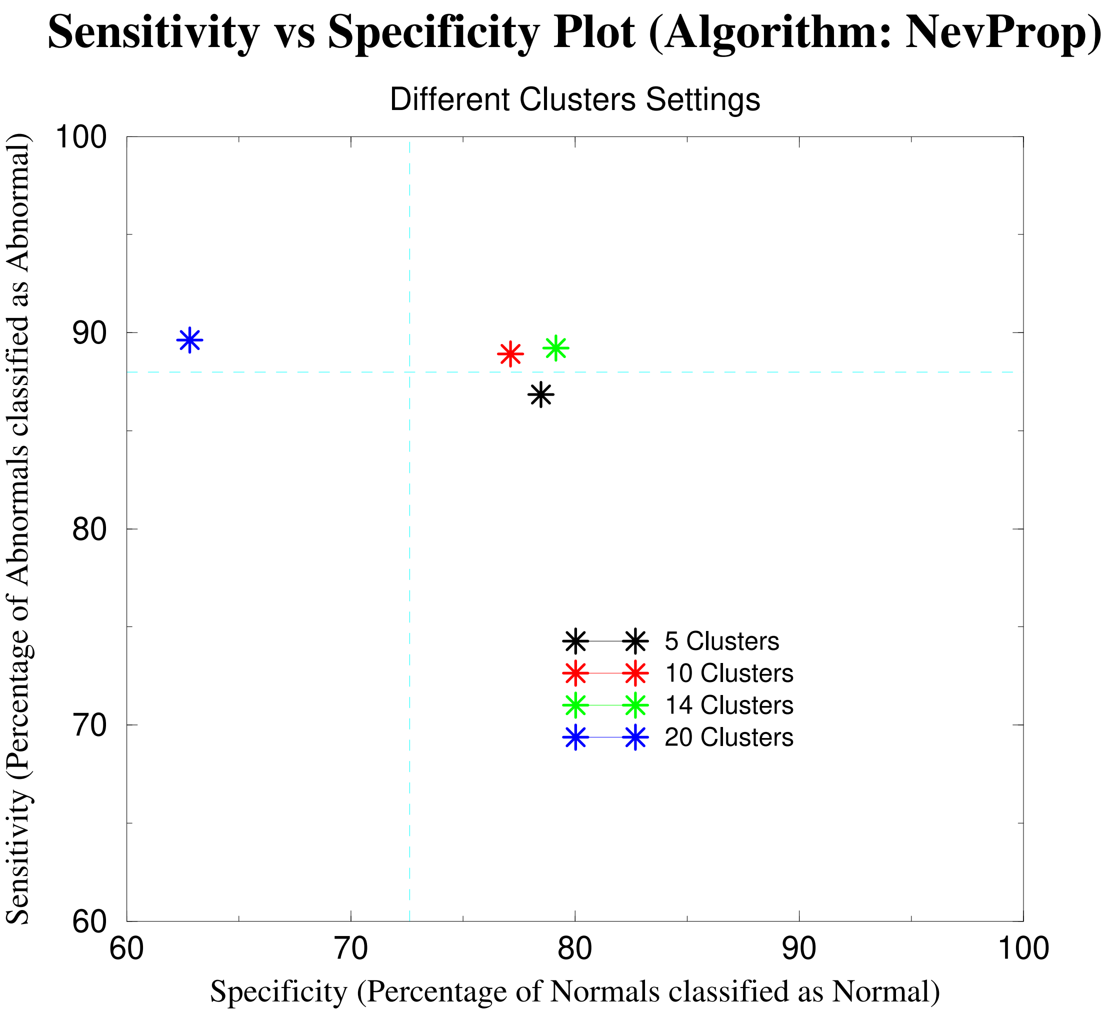
System Evaluation (NeoPath-1)

Sensitivity vs Specificity Plot (Algorithm: NevProp)



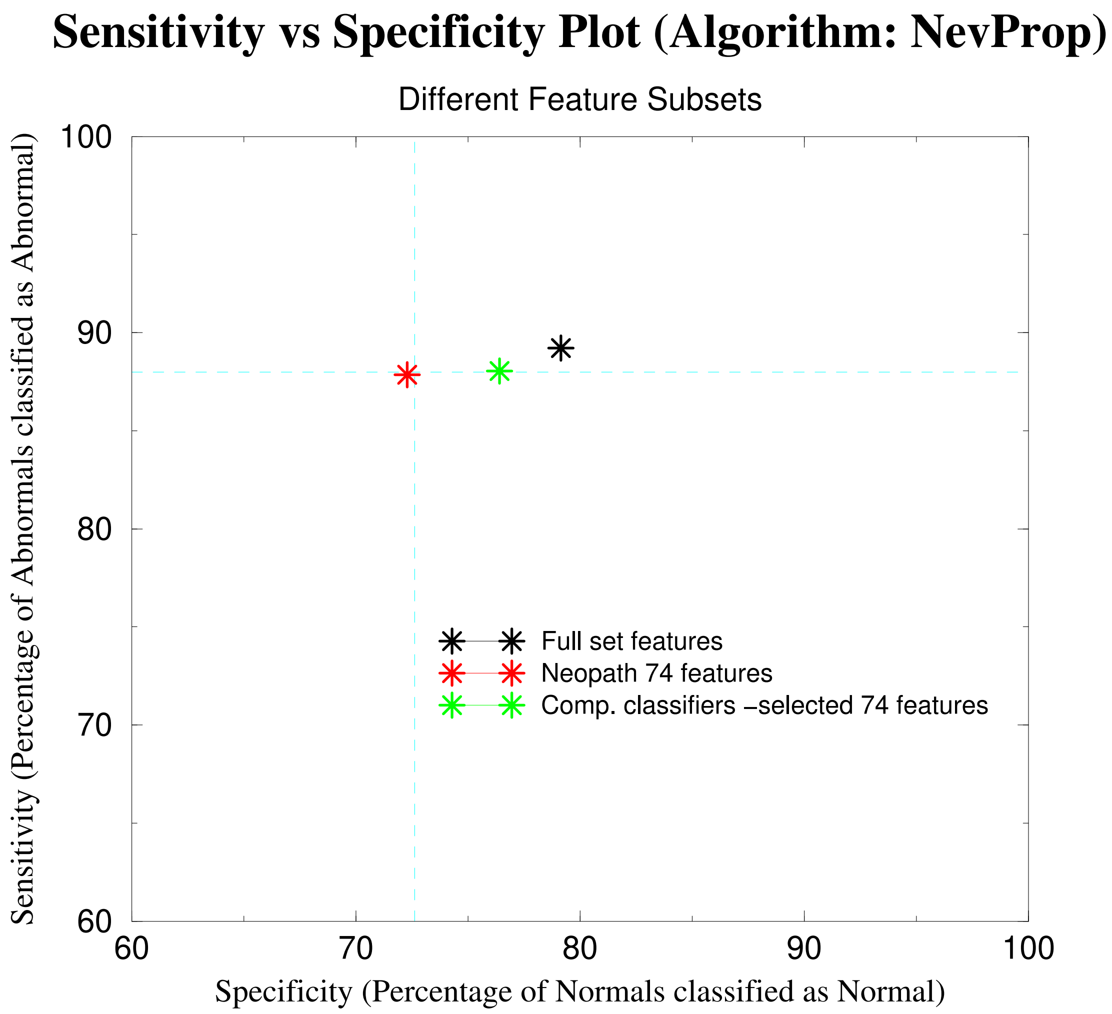
Various settings for target class and number of classifier.

System Evaluation (NeoPath-1)



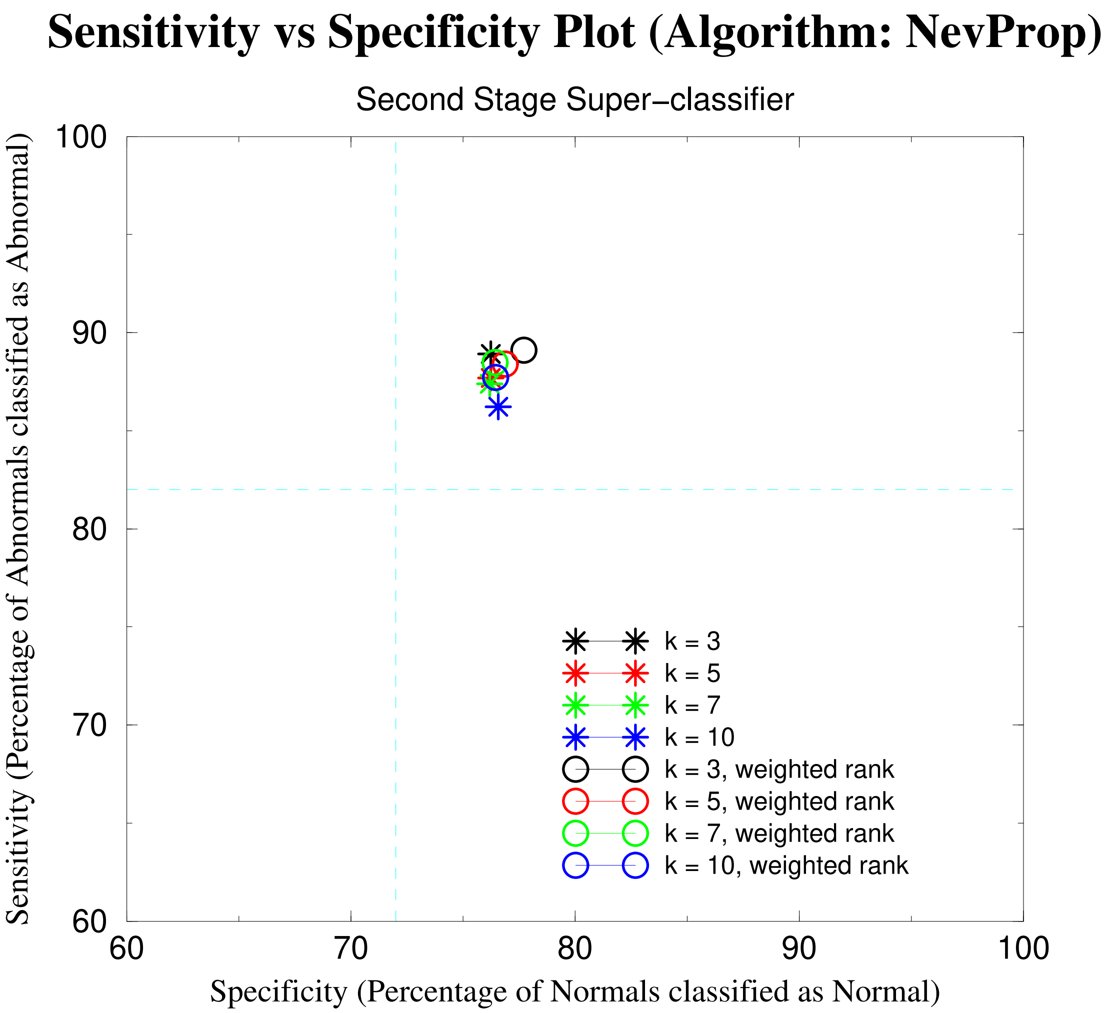
Various settings for different clusters.

System Evaluation (NeoPath-1)



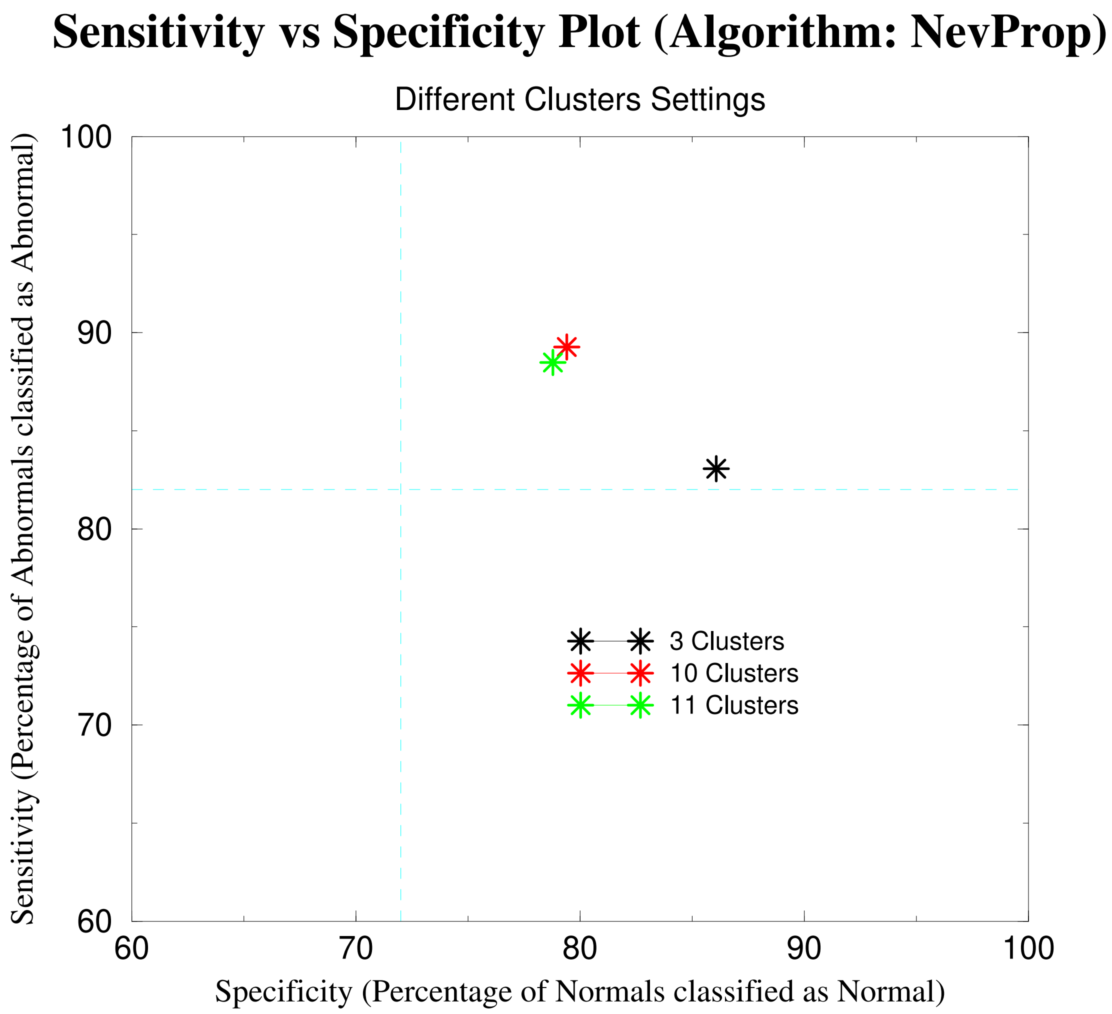
Various settings for different feature sets.

System Evaluation (NeoPath-2)



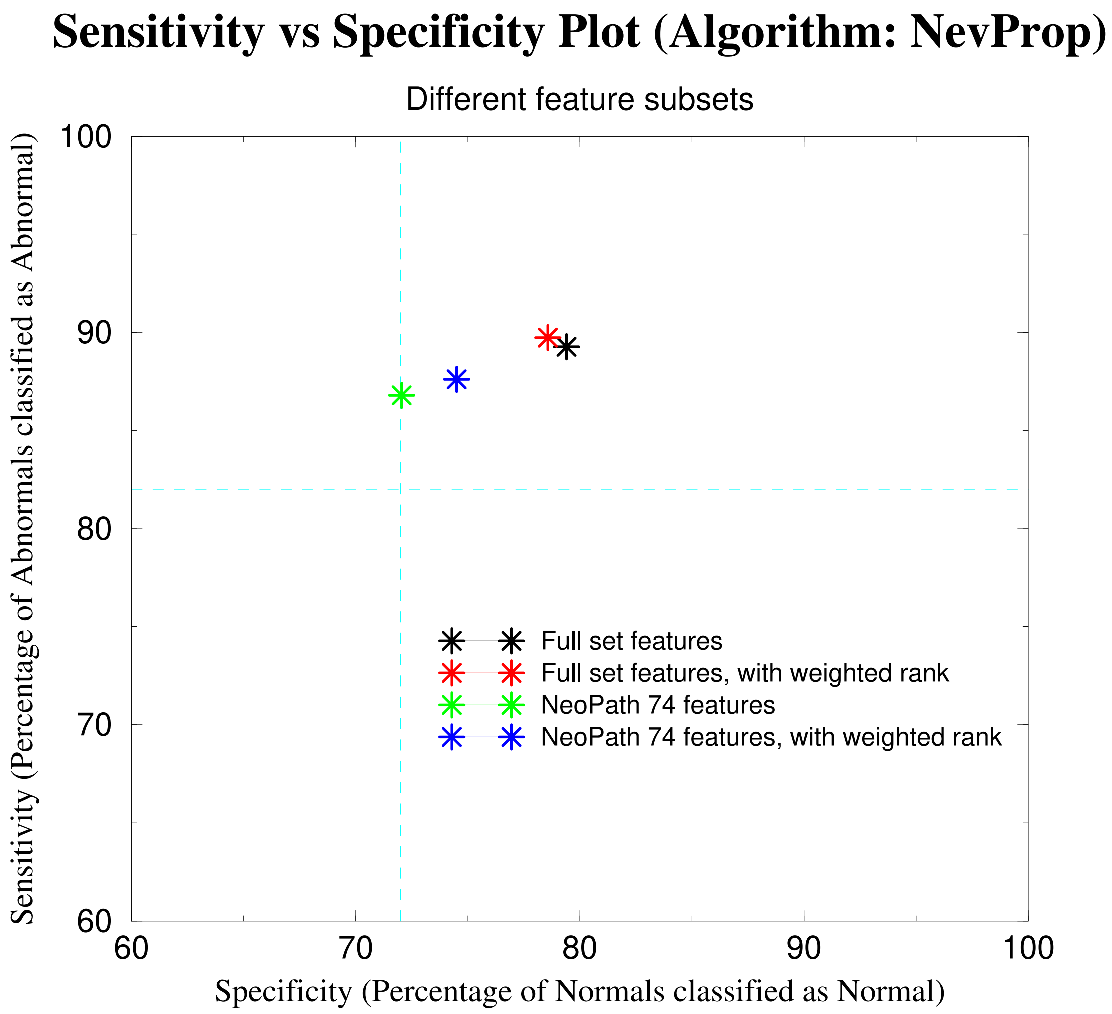
Various settings for target class and number of classifier.

System Evaluation (NeoPath-2)



Various settings for different clusters.

System Evaluation (NeoPath-2)



Various settings for different feature sets.

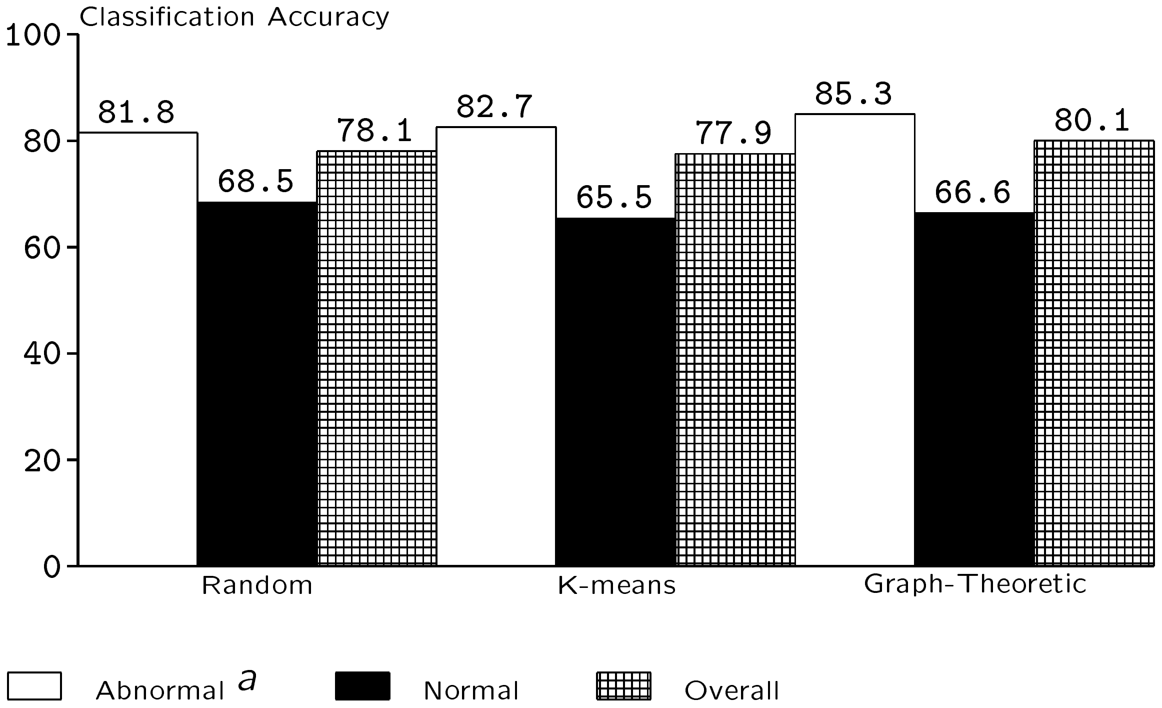
System Evaluation (Forest Cover Data)

- 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.

Algorithms	Accuracy %
Linear Discriminant Analysis	58
Backpropagation	70
NevProp	23.96
C4.5	63.64
NeuNet Pro SFAM	68 [a]
Hierarchical Multiple Classifier	70.81

[a] ≈ twice the number of training records than the other experiments.

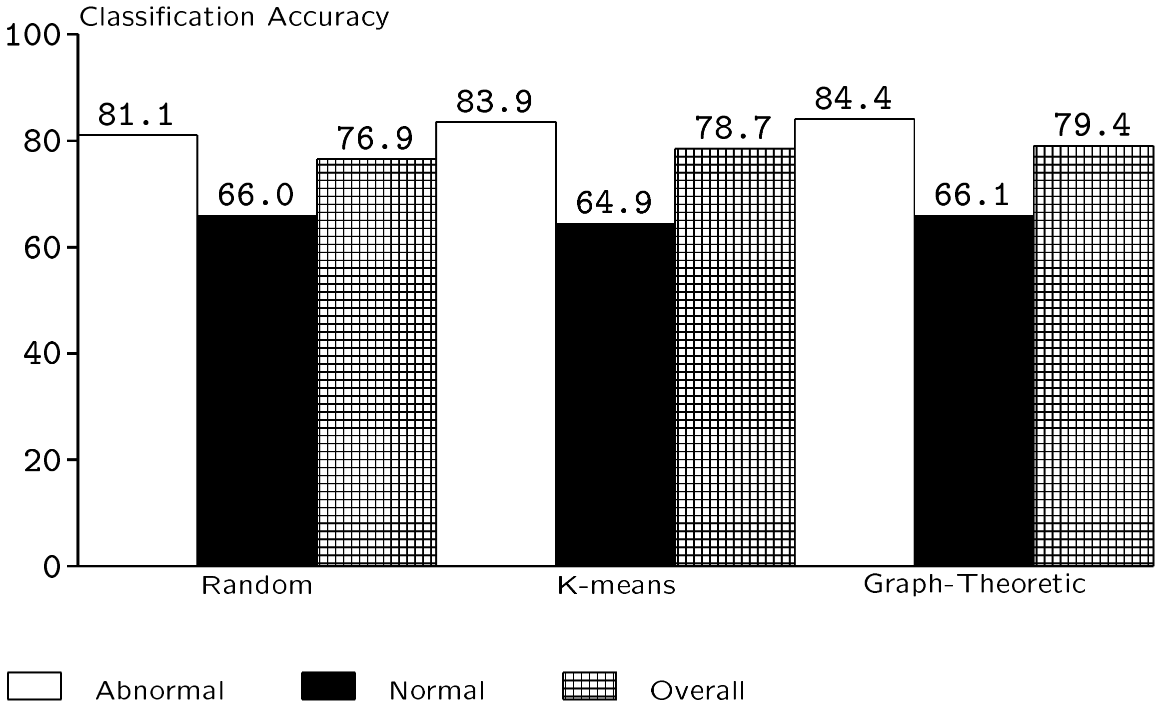
Comparison of Different Clustering Algorithms



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

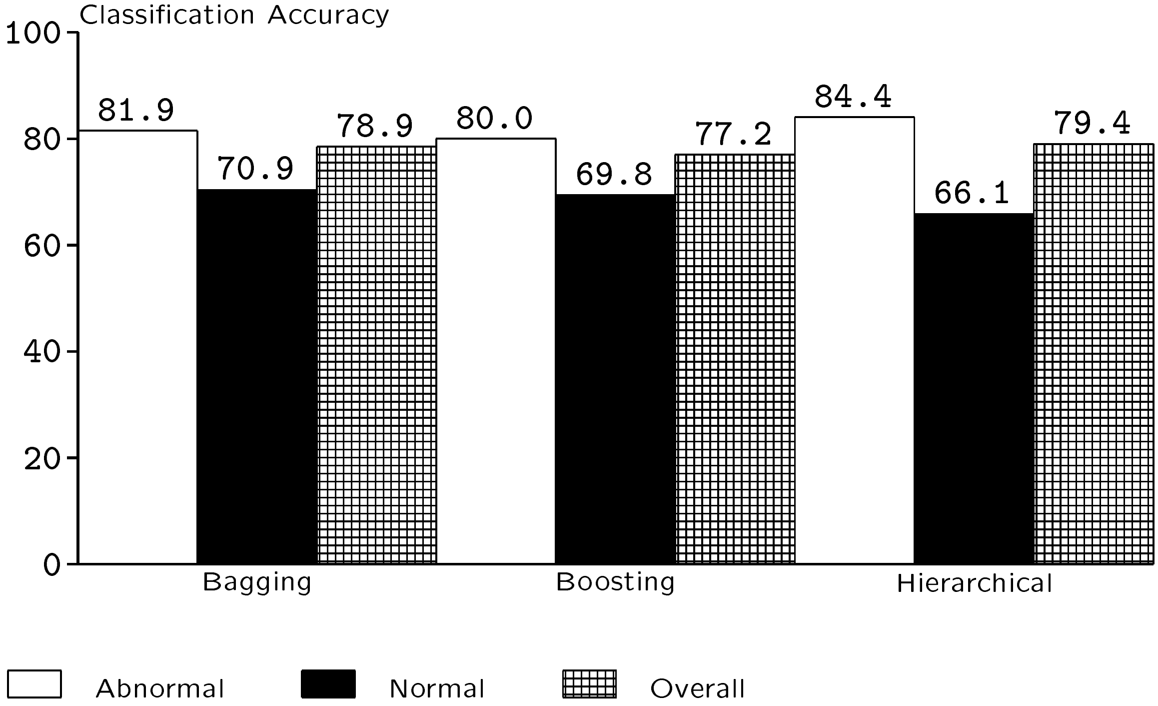
[a] The priority of identifying the abnormal cases is much higher than the normal cases.

Comparison of Different Clustering Algorithms



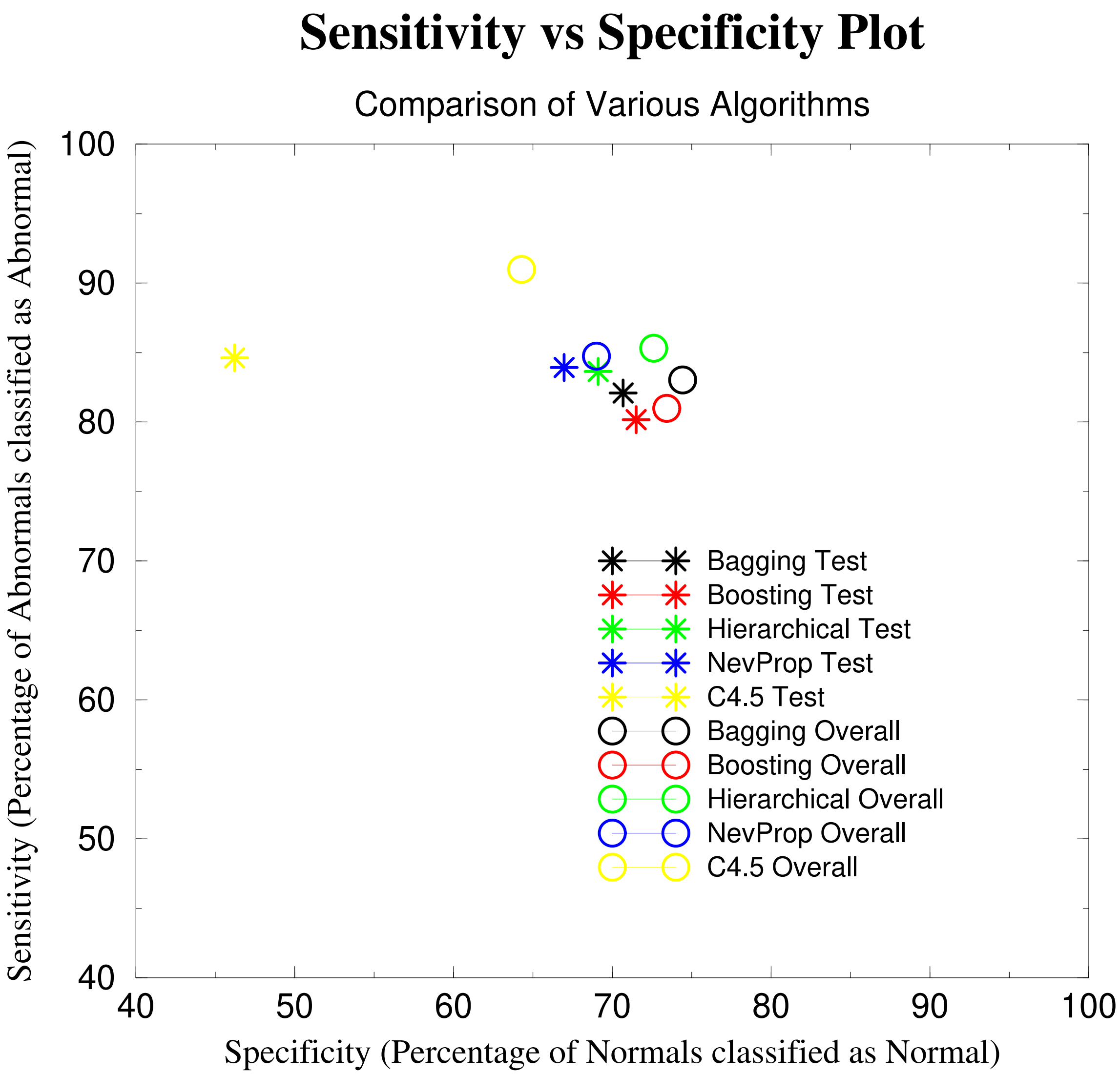
Classification accuracy of the NeoPath-2 test data with a subset of 74 features.

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-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 probabillities to be correctly classified.
- Adaption of other classification and clustering algorithms for different applications.