Multiple Classifiers for Multisource Remote Sensing Data

Jon Atli Benediktsson, Johannes R. Sveinsson and Gunnar J. Briem

Department of Electrical and Computer Engineering, University of Iceland

Outline

- Motivation
- Multiple Classifiers
  - Bagging
  - Boosting
  - Consensus Theoretic Classifiers
- Experimental Results
- Conclusion

Motivation

- Data fusion of multisource remote sensing and geographic data for classification purposes, has been an important research topic for more than a decade

Motivation

- Data fusion of multisource remote sensing and geographic data for classification purposes, has been an important research topic for more than a decade
- Different types of information from several data sources are used in order to improve the classification accuracy as compared to the accuracy achieved by single-source classification

Motivation

HÁSKÓLI ÍSLANDS

- Data fusion of multisource remote sensing and geographic data for classification purposes, has been an important research topic for more than a decade
- Different types of information from several data sources are used in order to improve the classification accuracy as compared to the accuracy achieved by single-source classification
- A major observation in previous research on multisource classification, is that conventional parametric statistical pattern recognition methods are not appropriate in classification of such data

Motivation

- Here, we are interested in the use of an ensemble of classifiers or *multiple classifiers* for classification of multisource data

Motivation

- Here, we are interested in the use of an ensemble of classifiers or *multiple classifiers* for classification of multisource data
- Traditionally, in pattern recognition, a single classifier is used to determine which class a given pattern belongs to

Motivation

- Traditionally, in pattern recognition, a single classifier is used to determine which class a given pattern belongs to
- However, in many cases, the classification accuracy can be improved by using an ensemble of classifiers in the classification

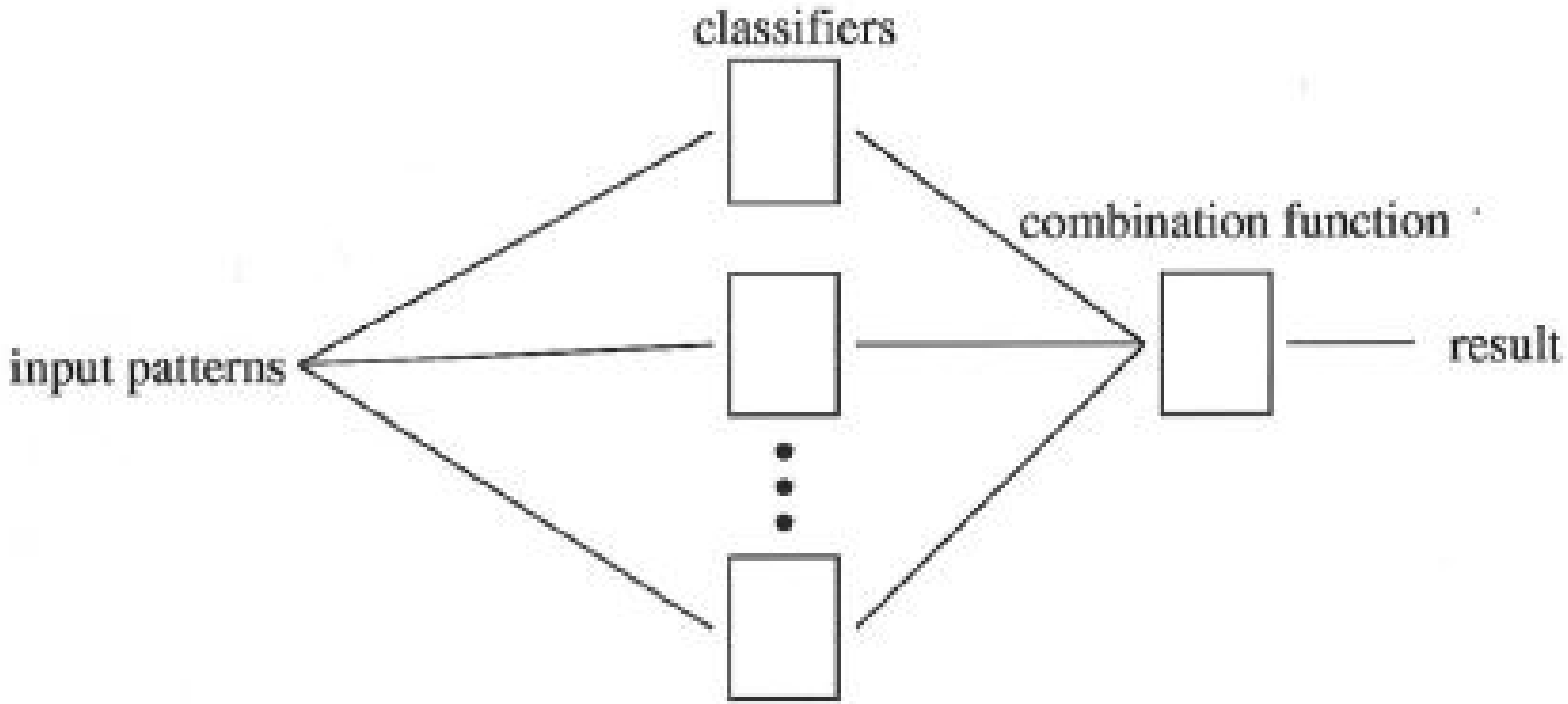
Motivation

- Traditionally, in pattern recognition, a single classifier is used to determine which class a given pattern belongs to
- However, in many cases, the classification accuracy can be improved by using an ensemble of classifiers in the classification
- In such cases it is possible to have the individual classifiers support each other in making a decision

Motivation

- Traditionally, in pattern recognition, a single classifier is used to determine which class a given pattern belongs to
- However, in many cases, the classification accuracy can be improved by using an ensemble of classifiers in the classification
- In such cases it is possible to have the individual classifiers support each other in making a decision
- The aim is to determine an effective combination method which makes use of the benefits of each classifier but avoids the weaknesses

Multiple Classifier – Diagram



A Schematic of a Multiple Classifier

Multiple Classifiers

- Three Approaches Considered:

Multiple Classifiers

- Three Approaches Considered:
  - Bagging

Multiple Classifiers

- Three Approaches Considered:
  - Bagging
  - Boosting

Multiple Classifiers

- Three Approaches Considered:
  - Bagging
  - Boosting
  - Consensus Theoretic Classifiers

Bagging

- Proposed in 1994 by Breiman
- Simple method
- m samples randomly and uniformly selected from a sample set of size m
- Done in parallel or series
- Uses resampling but not re-weighting
- All classifiers have equal weights
- Reduces classification variance

Bagging Algorithm:

Bagging Algorithm:

Input: A training set S with m samples, where each sample Z_j is from class ω_j, the base classifier is $\mathcal{I}$ and the number of bootstrapped sets is T.

1. For $i = 1$ to $T\{$
2. $\quad S_i =$ bootstrapped bag from S
3. $\quad C_i = \mathcal{I}(S_i)$
4. $\}$
5. $C^*(Z) = \underset{\omega \in \Omega}{\arg\max} \sum_{i:C_i(Z)=\omega} 1$

Output: The multiple classifier C^*.

Boosting

- Proposed in 1989 by Schapire

Boosting

- Proposed in 1989 by Schapire
- Potentially increases the accuracy of any base classifier

Boosting

- Proposed in 1989 by Schapire
- Potentially increases the accuracy of any base classifier
- AdaBoost proposed in 1995 by Freund and Schapire

Boosting

- AdaBoost proposed in 1995 by Freund and Schapire
  - Concentrates on difficult samples

Boosting

- AdaBoost proposed in 1995 by Freund and Schapire
  - Concentrates on difficult samples
  - Tends to not overfit noiseless data

Boosting

- AdaBoost proposed in 1995 by Freund and Schapire
  - Concentrates on difficult samples
  - Tends to not overfit noiseless data
  - Reduces classification variance and bias

Boosting

- AdaBoost proposed in 1995 by Freund and Schapire
  - Concentrates on difficult samples
  - Tends to not overfit noiseless data
  - Reduces classification variance and bias
  - Done in series

Boosting

- AdaBoost proposed in 1995 by Freund and Schapire
  - Concentrates on difficult samples
  - Tends to not overfit noiseless data
  - Reduces classification variance and bias
  - Done in series
  - Computationally demanding

Boosting

- AdaBoost proposed in 1995 by Freund and Schapire
  - Concentrates on difficult samples
  - Tends to not overfit noiseless data
  - Reduces classification variance and bias
  - Done in series
  - Computationally demanding
  - Bad performance on noisy data

Boosting

- AdaBoost proposed in 1995 by Freund and Schapire
  - Concentrates on difficult samples
  - Tends to not overfit noiseless data
  - Reduces classification variance and bias
  - Done in series
  - Computationally demanding
  - Bad performance on noisy data
  - Requires minimum accuracy of 0.5 for each base classifier

AdaBoost.M1 Algorithm:

Input: A training set S with m samples, where each sample Z_j is from class ω_j, the base classifier is $\mathcal{I}$ and the number of classifiers is T.

AdaBoost.M1 Algorithm:

1. $S_1 = S$ and weight$(Z_j) = 1$ for $j = 1 \dots m$ ($Z \in S_1$)
2. For $i = 1$ to T{
3. $C_i = \mathcal{I}(S_i)$
4. $\epsilon_i = \frac{1}{m} \sum_{Z_j \in S_i : C_i(Z_j) \neq \omega_j}$ weight(Z_j)
5. If $\epsilon_i > 0.5$, set S_i to a bootstrap sample from S with weight$(Z) = 1 \,\forall\, x \in S_i$ and goto step 3.
   If ϵ_i is still > 0.5 after 25 iterations, abort!
6. $\beta_i = \epsilon_i / (1 - \epsilon_i)$
7. For each $Z_j \in S_i${ if $C_i(Z_j) = \omega_j$ then
   weight$(Z_j) =$ weight$(Z_j) \cdot \beta_i$}.
8. Norm weights such that the total weight of S_i is m.
9. }

AdaBoost.M1 Algorithm:

Output: The multiple classifier C^*.

Consensus Theory

- Consensus theory involves procedures for combining single probability distributions to summarize estimates from multiple data sources (multiple experts).

Consensus Theory

- Consensus theory involves procedures for combining single probability distributions to summarize estimates from multiple data sources (multiple experts).
- The data from each source are at first classified into a number of source-specific data classes.

Consensus Theory

- Consensus theory involves procedures for combining single probability distributions to summarize estimates from multiple data sources (multiple experts).
- The data from each source are at first classified into a number of source-specific data classes.
- The information from the sources is then aggregated by a global membership function and the data are classified according to the usual maximum selection rule into a number of user-specified information classes.

Consensus Theory

- Consensus theory involves procedures for combining single probability distributions to summarize estimates from multiple data sources (multiple experts).
- The data from each source are at first classified into a number of source-specific data classes.
- The information from the sources is then aggregated by a global membership function and the data are classified according to the usual maximum selection rule into a number of user-specified information classes.
- The combination formula obtained in consensus theory is called a consensus rule.

Consensus Theory

- Linear opinion pool (LOP):

$$C_j(Z) = \sum_{i=1}^{n} \lambda_i p(\omega_j|z_i)$$

$Z = [z_1, \ldots, z_n]$ input data vector, $p(\omega_j|z_i)$ source-specific posterior probability and λ_i's ($i = 1, \ldots, n$) source-specific weights.

Consensus Theory

- Linear opinion pool (LOP):

$$C_j(Z) = \sum_{i=1}^{n} \lambda_i p(\omega_j|z_i)$$

$Z = [z_1, \ldots, z_n]$ input data vector, $p(\omega_j|z_i)$ source-specific posterior probability and λ_i's ($i = 1, \ldots, n$) source-specific weights.

- Each of the data sources needs to be modeled.

Consensus Theory

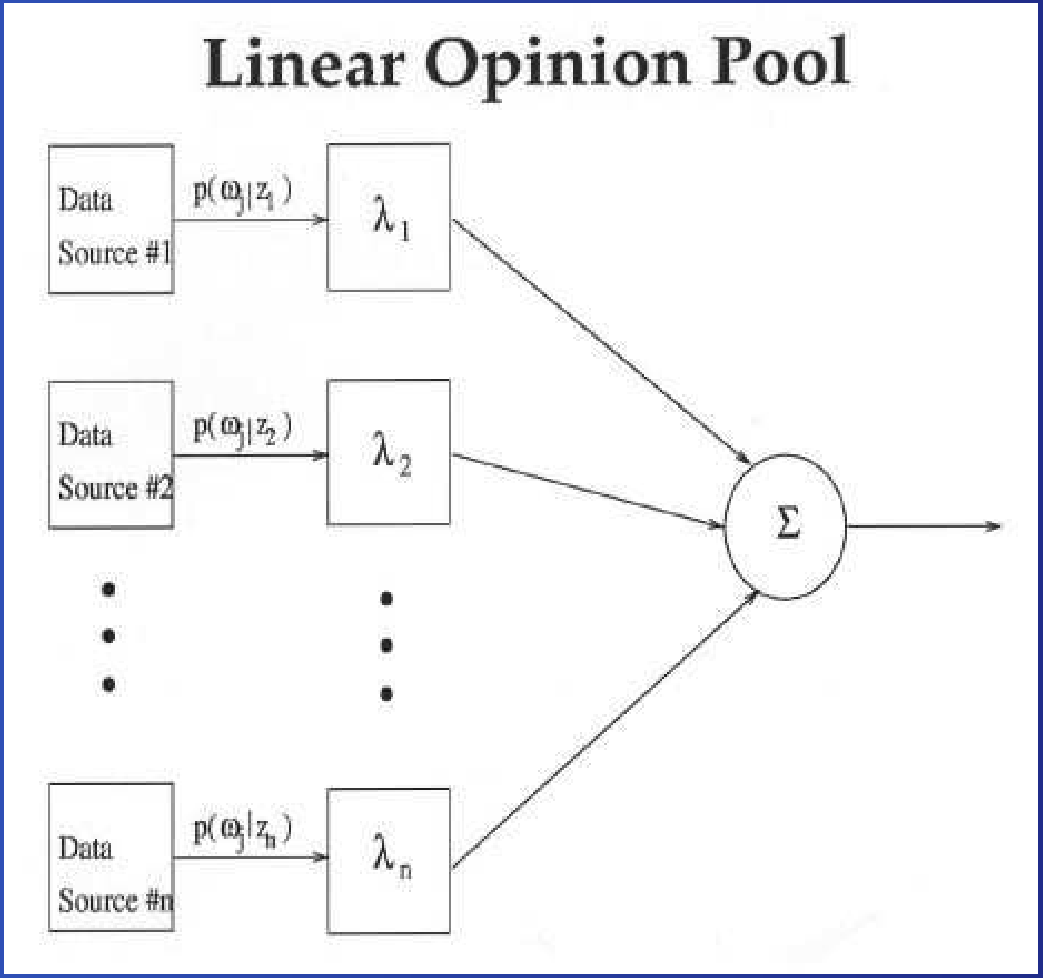
- Linear opinion pool (LOP):

$$C_j(Z) = \sum_{i=1}^{n} \lambda_i p(\omega_j|z_i)$$

$Z = [z_1, \ldots, z_n]$ input data vector, $p(\omega_j|z_i)$ source-specific posterior probability and λ_i's ($i = 1, \ldots, n$) source-specific weights.

- Each of the data sources needs to be modeled.
- The weights are associated with the sources in the global membership function to express quantitatively the goodness of each source.

Linear Opinion Pool



Consensus Rules

- Logarithmic opinion pool (LOGP):

$$L_j(Z) = \prod_{i=1}^{n} p(\omega_j|z_i)^{\lambda_i}$$

or

$$\log(L_j(Z)) = \sum_{i=1}^{n} \lambda_i \log(p(\omega_j|z_i)).$$

- $\lambda_1, \ldots, \lambda_n$ are weights which should reflect the goodness of the data sources.

Weighting of the Data Sources

HÁSKÓLI ÍSLANDS

- Each data source is at first treated separately and classified using statistical methods.

Weighting of the Data Sources

- Each data source is at first treated separately and classified using statistical methods.
- Weighting mechanisms are needed to control the influence of each data source in the combined classification.

Weighting of the Data Sources

- Each data source is at first treated separately and classified using statistical methods.
- Weighting mechanisms are needed to control the influence of each data source in the combined classification.
- The weights are optimized in order to improve the combined classification accuracies.

Weighting of the Data Sources

- Each data source is at first treated separately and classified using statistical methods.
- Weighting mechanisms are needed to control the influence of each data source in the combined classification.
- The weights are optimized in order to improve the combined classification accuracies.
- Both linear and non-linear methods are considered for the optimization.

Weighting

- The weight selection schemes in consensus theory should reflect the goodness of the separate input data, i.e., relatively high weights should be given to input data that contribute to high accuracy.

Weighting

- The weight selection schemes in consensus theory should reflect the goodness of the separate input data, i.e., relatively high weights should be given to input data that contribute to high accuracy.
- There are at least two potential weight selection schemes:

Weighting

- There are at least two potential weight selection schemes:
  - The first scheme is to select the weights such that they weight the individual sources but not the classes within the sources. One such possibility is to give all the sources equal weights (*equal weighting method*).

Weighting

- There are at least two potential weight selection schemes:
  - The first scheme is to select the weights such that they weight the individual sources but not the classes within the sources. One such possibility is to give all the sources equal weights (*equal weighting method*).
  - The second scheme is to choose the weights such that they not only weight the individual sources but also the classes within the sources. Here optimization can be performed (*optimal weighting method*).

Weighting

- In the second scheme, the combined output response, Y, can be written as

$$Y = f(X, \Lambda)$$

  where X contains source-specific posteriori discriminative information and Λ corresponds to the source-specific weights

- If f is linear and $Y = D$ is the desired output of the classifier then it is needed to solve

$$X\Lambda = D$$

  where Λ is an unknown matrix.

Weighting

- Λ's least square estimate, Λ_{lopt}, is sought to minimize the squared error, i.e.

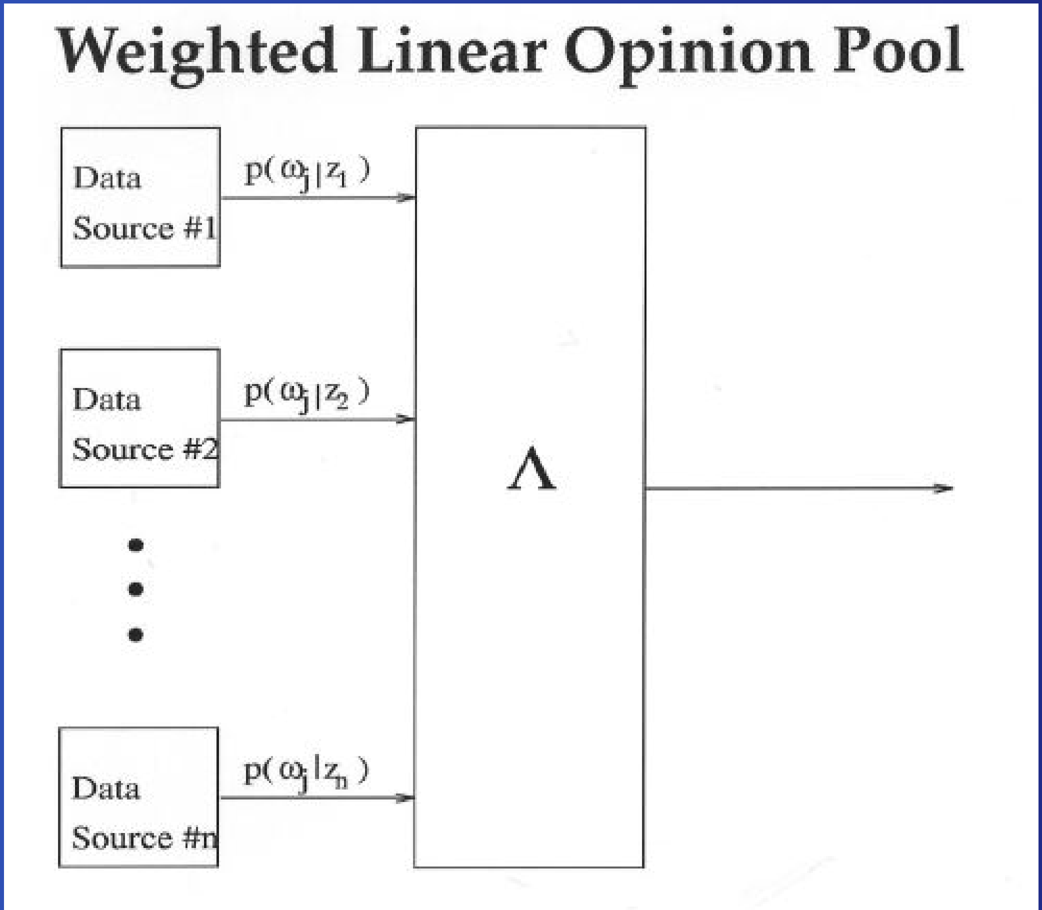
$$\Lambda_{lopt} = \arg \min_{\Lambda} \|X\Lambda - D\|^2.$$

- Λ_{lopt} is then given by

$$\Lambda_{lopt} = (X^T X)^{-1} X^T D$$

where X^T is the transpose of X, and $(X^T X)^{-1} X^T$ is the pseudo-inverse of X if $X^T X$ is non-singular.

Weighted LOP



Weighting

- When f in $Y = f(X, \Lambda)$ is non-linear:
  - A neural network or a genetic algorithm can be used to obtain an estimate of f.
  - The individual source classifiers can be considered to preprocess the data for the neural networks or the genetic algorithms.

Weighting

- A neural network or a genetic algorithm can be used to obtain an estimate of f.
- The individual source classifiers can be considered to preprocess the data for the neural networks or the genetic algorithms.
- If $Y = D$ is the desired output, the process can be described by

$$\Lambda_{nlopt} = \arg\min_{\Lambda} \|D - f(X, \Lambda)\|^2.$$

Weighting

- If $Y = D$ is the desired output, the process can be described by

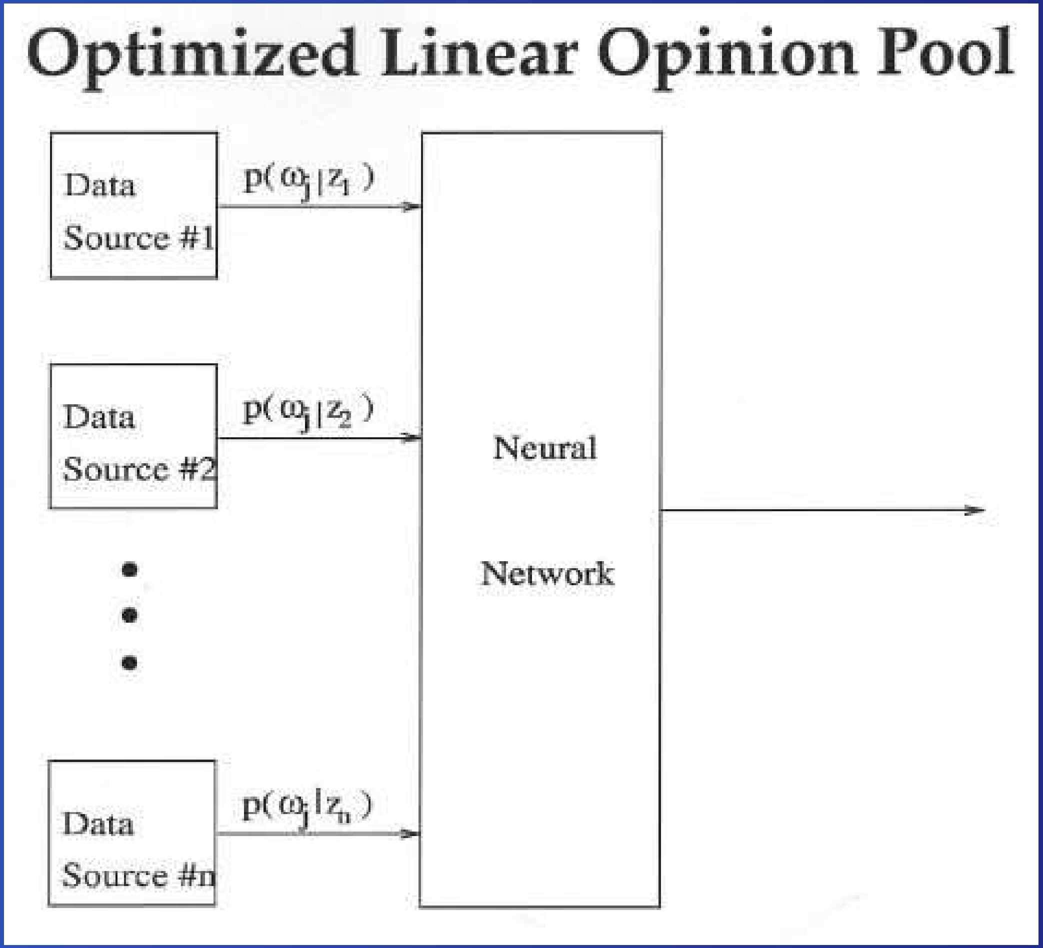
$$\Lambda_{nlopt} = \arg\min_{\Lambda} \|D - f(X, \Lambda)\|^2.$$

- The update equation for the weights of the neural network is

$$\Delta\Lambda_{nlopt} = \eta\|D - f(X, \Lambda)\|\nabla_{\Lambda} f$$

where η is a learning rate.

Optimized LOP



Experiments

- Two multisource geographic and remote sensing data sets were classified
  - Colorado Data Set
  - Anderson River Data Set
- Bagging and boosting was performed in Waikato Environment for Knowledge Analysis (WEKA)

Classifiers

- Minimum Euclidean Distance (MED)
- Maximum Likelihood (ML)
- Conjugate-Gradient Backpropagation (CGBP)
- Linear Opinion Pool (LOP)
- Logarithmic Opinion Pool (LOGP)
- Bagging
- Boosting (AdaBoost.M1)

Classifiers

- Bagging
- Boosting (AdaBoost.M1)
- Decision Table
- j4.8 (An implementation of the C4.5 decision tree)
- 1R (Classification based on one feature)

Colorado Data Set

- Classification was performed on a data set consisting of the following 4 data sources:
  - Landsat MSS data (4 spectral data channels).
  - Elevation data (in 10 m contour intervals, 1 data channel).
  - Slope data (0-90 degrees in 1 degree increments, 1 data channel).
  - Aspect data (1-180 degrees in 1 degree increments, 1 data channel).

Colorado Data Set

Class #	Information Class	Training Size	Test Size
1	Water	301	302
2	Colorado Blue Spruce	56	56
3	Mountane/Subalpine Meadow	43	44
4	Aspen	70	70
5	Ponderosa Pine 1	157	157
6	Ponderosa Pine/Douglas Fir	122	122
7	Engelmann Spruce	147	147
8	Douglas Fir/White Fir	38	38
9	Douglas Fir/Ponderosa Pine/Aspen	25	25
10	Douglas Fir/White Fir/Aspen	49	50
Total		1008	1011

Colorado Data Set – Training

Training Accuracies in Percentage for the Different Classification Methods Applied to the Colorado Data Set

Method	Cl. 1	Cl. 2	Cl. 3	Cl. 4	Cl. 5	Cl. 6	Cl. 7	Cl. 8	Cl. 9	Cl. 10	Avg. Acc.	Overall Accuracy
MED	41.5	98.2	25.6	37.1	37.6	0.0	73.5	0.0	40.0	24.5	**37.8**	**40.3**
Decision Table	100.0	89.3	67.4	84.3	54.1	80.3	100.0	36.8	24.0	93.9	**73.0**	**82.8**
j4.8	100.0	83.9	79.1	87.1	68.8	88.5	99.3	57.9	52.0	95.9	**81.3**	**88.0**
1R	100.0	0.0	0.0	0.0	38.2	63.1	97.3	23.7	0.0	36.7	**35.9**	**60.3**
CGBP (40 hidden neurons)	100.0	96.7	95.7	99.5	90.3	89.5	100.0	87.3	96.7	100.0	**95.6**	**96.3**
LOP (equal weights)	100.0	0.0	0.0	92.9	38.9	49.2	100.0	0.0	12.0	100.0	**49.3**	**68.1**
LOP (heuristic weights)	100.0	25.0	16.3	91.4	36.3	90.2	99.3	0.0	0.0	100.0	**55.8**	**74.2**
LOP (optimal linear weights)	100.0	62.5	25.6	74.3	66.2	79.5	98.6	23.7	40.0	91.8	**66.2**	**80.3**
LOP (optimized with CGBP)	100.0	87.9	26.2	81.8	67.8	73.0	100.0	39.5	75.0	94.4	**74.6**	**83.5**
LOGP (equal weights)	99.7	96.4	20.9	87.1	60.5	46.7	100.0	44.7	44.0	91.8	**69.2**	**79.0**
LOGP (heuristic weights)	99.7	91.1	23.3	95.7	45.2	83.6	100.0	5.3	48.0	100.0	**69.2**	**80.5**
LOGP (optimal linear weights)	100.0	67.9	23.3	81.4	58.6	82.8	98.6	18.4	28.0	91.8	**65.1**	**79.7**
LOGP (optimized with CGBP)	100.0	80.4	69.8	99.6	78.3	82.8	100.0	80.3	100.0	100.0	**89.1**	**91.4**
Bagging (Decision Table)	100.0	69.6	76.7	95.7	81.5	82.0	100.0	31.6	60.0	98.0	**79.5**	**88.3**
Bagging (j4.8)	100.0	89.3	74.4	92.9	84.1	81.1	100.0	55.3	72.0	93.9	**84.3**	**90.4**
Bagging (1R)	100.0	41.1	60.5	61.4	50.3	68.0	97.3	21.1	32.0	83.7	**61.5**	**74.9**
Boosting (Decision Table)	100.0	82.1	88.4	98.6	75.2	83.6	100.0	68.4	100.0	100.0	**89.6**	**91.4**
Boosting (j4.8)	100.0	98.2	97.7	100.0	91.1	94.3	100.0	94.7	100.0	100.0	**97.6**	**97.5**
Boosting (1R)	100.0	94.6	79.1	95.7	78.3	76.2	100.0	63.2	100.0	100.0	**88.7**	**90.9**
Number of Samples	301	56	43	70	157	122	147	38	25	49		1008

Colorado Data Set – Test

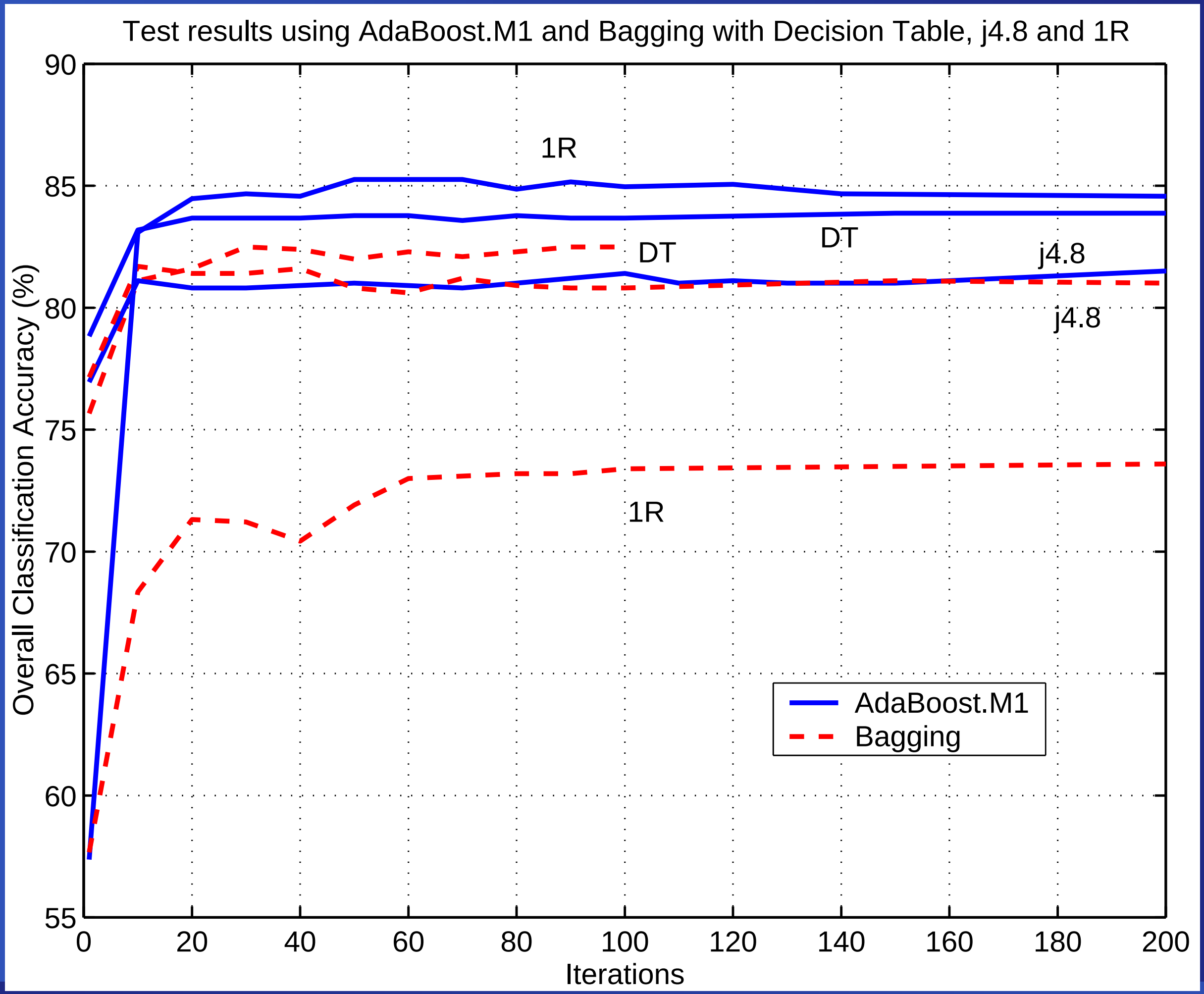
Test Accuracies in Percentage for the Different Classification Methods Applied to the Colorado Data Set

Method	Cl. 1	Cl. 2	Cl. 3	Cl. 4	Cl. 5	Cl. 6	Cl. 7	Cl. 8	Cl. 9	Cl. 10	Avg. Acc.	Overall Accuracy
MED	40.1	100.0	34.1	30.0	32.5	0.8	69.4	0.0	28.0	20.0	**35.5**	**38.0**
Decision Table	100.0	80.4	50.0	74.3	47.1	73.0	96.6	28.9	4.0	80.0	**63.4**	**77.0**
j4.8	100.0	62.5	54.5	74.3	57.3	66.4	98.0	28.9	8.0	84.0	**63.4**	**77.4**
1R	100.0	0.0	0.0	0.0	30.6	65.6	95.9	15.8	0.0	24.0	**33.2**	**58.3**
CGBP (40 hidden neurons)	99.9	57.1	61.0	67.6	59.3	69.1	97.4	34.6	45.3	78.7	**67.0**	**78.4**
LOP (equal weights)	100.0	0.0	0.0	87.1	35.0	48.4	100.0	0.0	0.0	94.0	**46.5**	**66.4**
LOP (heuristic weights)	100.0	30.4	18.2	80.0	35.7	88.5	100.0	0.0	0.0	96.0	**54.9**	**73.4**
LOP (optimal linear weights)	100.0	80.4	25.0	77.1	66.3	75.4	99.3	15.8	32.0	92.0	**66.1**	**80.2**
LOP (optimized with CGBP)	100.0	90.2	39.2	75.3	61.0	74.6	99.3	34.9	58.0	96.5	**72.9**	**82.2**
LOGP (equal weights)	99.3	100.0	18.2	85.7	56.7	52.5	99.3	42.1	44.0	92.0	**69.0**	**78.7**
LOGP (heuristic weights)	100.0	96.4	18.2	91.4	40.8	87.7	99.3	10.5	24.0	100.0	**66.8**	**79.6**
LOGP (optimal linear weights)	100.0	76.8	25.0	75.7	63.7	81.1	99.3	13.2	16.0	92.0	**64.3**	**80.0**
LOGP (optimized with CGBP)	99.8	64.3	58.0	73.9	61.5	71.7	98.6	49.3	80.0	94.0	**75.1**	**82.3**
Bagging (Decision Table)	100.0	66.1	72.7	80.0	73.2	72.1	99.3	15.8	20.0	94.0	**69.3**	**82.5**
Bagging (j4.8)	100.0	60.7	63.6	75.7	69.4	75.4	99.3	28.9	40.0	82.0	**69.5**	**81.7**
Bagging (1R)	100.0	42.9	54.5	61.4	44.6	70.5	98.0	13.2	24.0	80.0	**58.9**	**73.6**
Boosting (Decision Table)	100.0	67.9	70.5	80.0	67.5	73.0	99.3	28.9	76.0	98.0	**76.1**	**83.8**
Boosting (j4.8)	100.0	60.7	70.5	77.1	65.6	67.2	98.6	42.1	60.0	84.0	**72.6**	**81.5**
Boosting (1R)	100.0	73.2	70.5	77.1	68.2	73.0	100.0	60.5	72.0	100.0	**79.4**	**85.3**
Number of Samples	302	56	44	70	157	122	147	38	25	50		1011

Colorado Results

- All multiple classifier schemes show improvement over single classifiers
- Highest training accuracies using AdaBoost on j4.8

Colorado – Test accuracies



Anderson River Data Set

- Six data sources were used:
  - Airborne Multispectral Scanner (AMSS) with 11 spectral data channels (10 channels from 380 to 1100 nm and 1 channel from 8 to 14 μm).
  - Steep Mode Synthetic Aperture Radar (SAR) with 4 data channels (X-HH, X-HV, L-HH, L-HV).
  - Shallow Mode SAR with 4 data channels (X-HH, X-HV, L-HH, L-HV).
  - Elevation data (1 data channel, where elevation in meters = 61.996 + 7.2266 * pixel value).
  - Slope data (1 data channel, where slope in degrees = pixel value).

Anderson River – Data Set

Class #	Information Class	Training Size	Test Size
1	Douglas Fir (31-40m)	971	1250
2	Douglas Fir (21-30m)	551	817
3	Douglas Fir + Other Species(31-40m)	548	701
4	Douglas Fir + Lodgepole Pine (21-30m)	542	705
5	Hemlock + Cedar (31-40m)	317	405
6	Forest Clearings	1260	1625
Total		4189	5503

Anderson River – Training

Training Accuracies in Percentage for the Different Classification Methods Applied to the Anderson River Data Set

Method	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Average Accuracy	Overall Accuracy
MED	40.4	8.9	47.6	67.7	42.3	72.4	**46.6**	**50.5**
ML	54.6	31.6	87.8	90.9	81.4	73.3	**69.9**	**68.2**
Decision Table	78.7	59.3	76.8	70.8	75.7	83.4	**74.1**	**76.1**
j4.8	93.9	91.5	93.8	93.9	96.2	97.0	**94.4**	**94.7**
1R	61.3	6.5	22.8	74.2	19.2	77.5	**43.6**	**52.4**
CGBP (30 hidden neurons)	72.2	34.4	67.2	74.6	79.2	83.1	**68.4**	**70.7**
LOP (equal weights)	49.6	0.0	0.0	51.5	0.0	94.9	**32.7**	**47.6**
LOP (heuristic weights)	68.2	0.0	0.0	73.1	24.3	89.4	**42.5**	**54.0**
LOP (optimal linear weights)	69.8	42.7	81.20	77.5	70.4	78.9	**70.1**	**71.5**
LOP (optimized with CGBP)	69.0	45.0	81.3	76.9	85.0	78.4	**72.6**	**71.8**
LOGP (equal weights)	68.7	28.1	79.6	78.8	81.7	74.3	**68.5**	**68.8**
LOGP (heuristic weights)	68.9	33.2	78.5	79.5	75.7	75.8	**68.6**	**69.4**
LOGP (optimal linear weights)	71.9	40.3	79.7	75.1	82.0	79.1	**71.4**	**72.1**
LOGP (optimized with CGBP)	81.2	56.0	84.3	88.7	91.7	86.4	**81.4**	**81.6**
Bagging (Decision Table)	97.6	91.5	97.6	96.9	100.0	99.0	**97.1**	**97.3**
Bagging (j4.8)	98.7	96.2	97.4	98.3	99.4	99.3	**98.2**	**98.4**
Bagging (1R)	75.1	14.9	46.7	48.7	71.6	80.6	**56.3**	**61.4**
Boosting (Decision Table)	99.5	97.3	99.1	99.3	99.4	99.7	**99.0**	**99.2**
Boosting (j4.8)	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**	**100.0**
Boosting (1R)	84.7	71.9	85.0	90.2	96.8	93.3	**87.0**	**87.3**
Number of Samples	971	551	548	542	317	1260		4189

Anderson River – Test

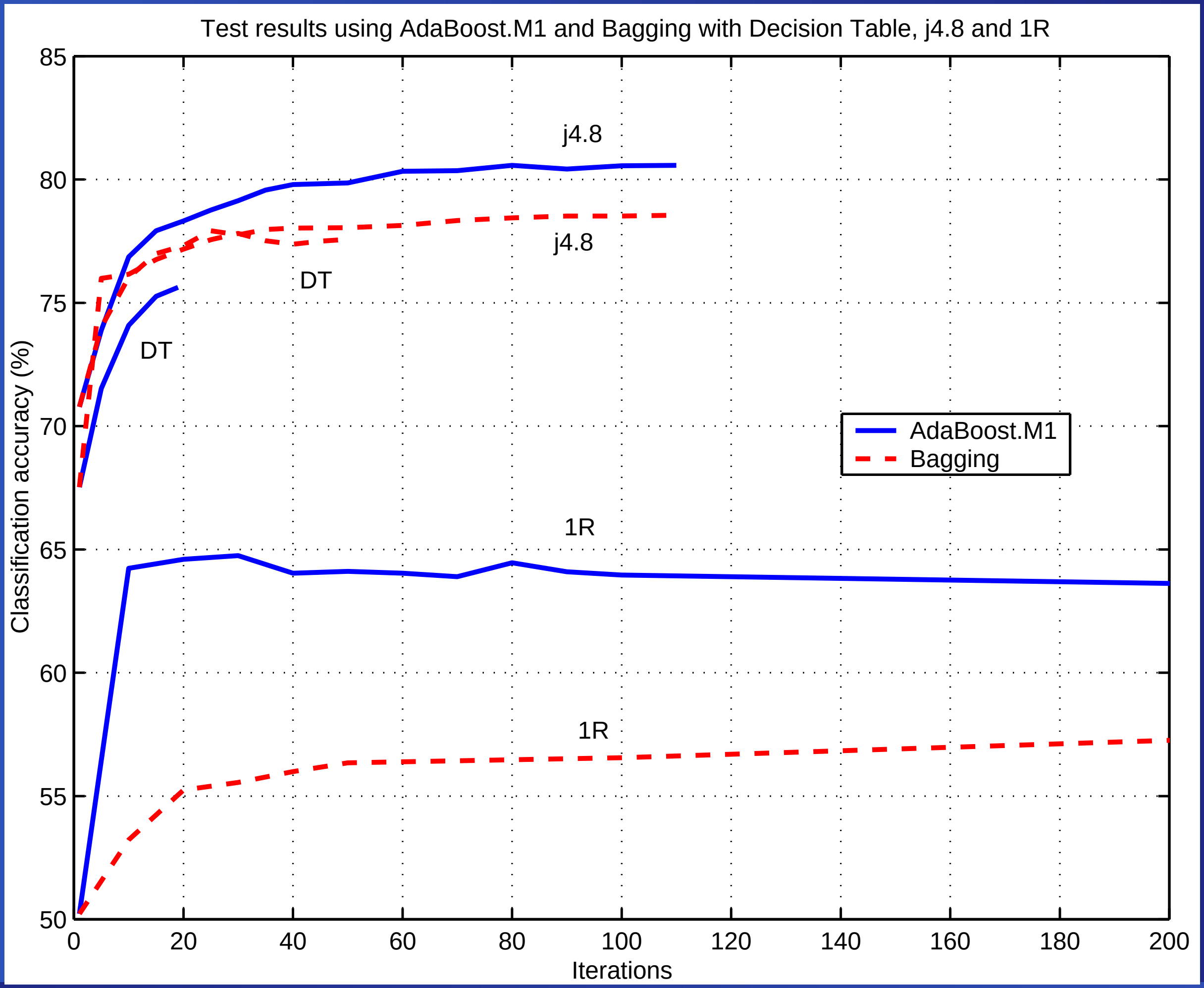
Test Accuracies in Percentage for the Different Classification Methods Applied to the Anderson River Data Set

Method	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Average Accuracy	Overall Accuracy
MED	39.7	8.9	48.4	70.2	46.0	71.7	**47.5**	**50.8**
ML	50.8	27.7	84.5	81.9	73.8	72.0	**64.3**	**65.1**
Decision Table	73.8	42.4	66.5	61.7	72.8	77.0	**65.7**	**67.5**
j4.8	71.2	47.4	69.2	72.3	74.8	81.2	**69.4**	**70.8**
1R	58.5	4.3	19.3	74.5	20.0	77.4	**42.3**	**50.2**
CGBP (30 hidden neurons)	71.9	29.3	67.5	73.8	79.3	82.4	**67.4**	**68.8**
LOP (equal weights)	49.8	0.0	0.0	50.4	0.0	95.3	**32.6**	**45.8**
LOP (heuristic weights)	68.9	0.0	0.0	73.1	20.8	89.3	**42.0**	**53.9**
LOP (optimal linear weights)	66.4	34.3	78.5	74.8	72.6	79.5	**67.7**	**68.6**
LOP (optimized with CGBP)	67.1	36.7	77.3	75.1	83.4	77.6	**69.5**	**69.2**
LOGP (equal weights)	67.9	23.1	77.8	77.5	81.2	73.7	**66.9**	**66.4**
LOGP (heuristic weights)	69.0	31.8	75.9	78.6	75.6	75.1	**67.6**	**68.6**
LOGP (optimal linear weights)	68.6	32.4	75.2	71.2	81.7	80.1	**68.2**	**68.7**
LOGP (optimized with CGBP)	75.4	43.1	76.9	79.5	87.2	82.1	**74.0**	**74.1**
Bagging (Decision Table)	80.7	48.2	82.9	77.3	89.6	85.5	**77.4**	**77.8**
Bagging (j4.8)	80.0	51.2	81.3	79.6	86.4	87.5	**77.7**	**78.5**
Bagging (1R)	72.7	11.6	39.7	48.5	70.9	80.7	**54.0**	**58.6**
Boosting (Decision Table)	77.3	51.7	73.9	75.0	83.7	85.4	**74.5**	**75.6**
Boosting (j4.8)	83.0	54.2	81.9	81.4	88.9	88.9	**79.7**	**80.6**
Boosting (1R)	61.1	36.2	58.3	69.5	67.4	81.9	**62.4**	**64.7**
Number of Samples	1250	817	701	705	405	1625		5503

Anderson River Results

- Bagging with j4.8 is more accurate than consensus theoretic classification
- AdaBoost gives higher accuracies than bagging

Anderson River Results



Conclusion

- Multiple classification generally improves on single classification in terms of accuracies
- Adaboost with well selected base classifiers was the most accurate method in experiments