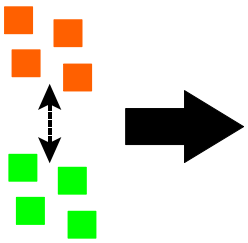
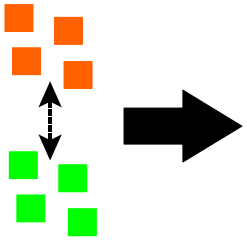
Discrimination Among Groups



- Are groups significantly different? (How valid are the groups?)
  - Multivariate Analysis of Variance [(NP)MANOVA]
  - Multi-Response Permutation Procedures [MRPP]
  - Analysis of Group Similarities [ANOSIM]
  - Mantel's Test [MANTEL]

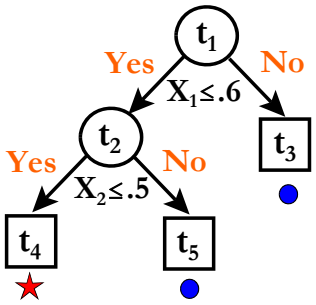


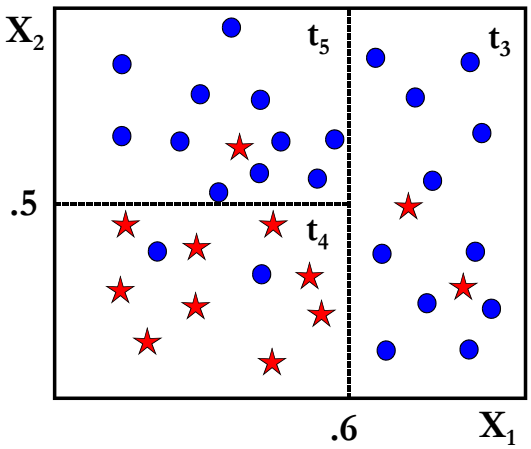
- How do groups differ? (Which variables best distinguish among the groups?)
  - Discriminant Analysis [DA]
  - Classification and Regression Trees [CART]
  - Logistic Regression [LR]
  - Indicator Species Analysis [ISA]

Classification (and Regression) Trees

- *Nonparametric* procedure useful for exploration, description, and prediction of grouped data (Breiman et al. 1998; De'ath and Fabricius 2000).

- Recursive *partitioning* of the data space such that the 'populations' within each partition become more and more class homogeneous.





Important Characteristics of CART

- Requires specification of only a *few elements*:
  - A set of questions of the form: Is $x_m \leq c$?
  - A rule for selecting the best split at any node.
  - A criterion for choosing the right-sized tree.
- Can be applied to *any data structure*, including mixed data sets containing both continuous, categorical, and count variables, and both standard and nonstandard data structures.
- Can handle *missing data*; no need to drop observations with a missing value on one of the variables.
- Final classification has a *simple form* which can be compactly stored and that efficiently classifies new data.

Important Characteristics of CART

- Makes powerful use of conditional information in handling *nonhomogeneous* relationships.
- Automatic stepwise *variable selection* and complexity reduction.
- Provides not only a classification, but also an estimate of the *misclassification probability* for each object.
- In a standard data set it is invariant under all monotonic *transformations* of continuous variables.
- Extremely robust with respect to *outliers* and misclassified points.
- Tree output gives easily *understood* and *interpreted* information regarding the predictive structure of the data.

Important Characteristics of CART

- Capable of exploring *complex data sets* not easily handled by other techniques, where complexity can include:
  - High dimensionality
  - Mixture of data types
  - Nonstandard data structure
  - Nonhomogeneity

"*The curse of dimensionality*" (Bellman 1961)

"Ten points on the unit interval are not distant neighbors. But 10 points on a 10-dimensional unit rectangle are like oases in the desert." (Breiman et al. 1998)

High dimensionality is not a problem for CART!

Important Characteristics of CART

- Capable of exploring *complex data sets* not easily handled by other techniques, where complexity can include:
  - High dimensionality
  - Mixture of data types
  - Nonstandard data structure
  - Nonhomogeneity

Ecological data sets commonly include a mixture of categorical and continuous variables.

- Cover type (categorical)
- Patch size (continuous)
- Number of wetlands w/i 1 km (count)

A mixture of data types is not a problem for CART!

Important Characteristics of CART

- Capable of exploring *complex data sets* not easily handled by other techniques, where complexity can include:
  - High dimensionality
  - Mixture of data types
  - Nonstandard data structure
  - Nonhomogeneity

Standard Structure

	X_1	X_2	X_3	X_4	X_5	X_6
1	x	x	x	x	x	x
2	x	x	x	x	x	x
3	x	x	x	x	x	x
4	x	x	x	x	x	x
5	x	x	x	x	x	x

Nonstandard Structure

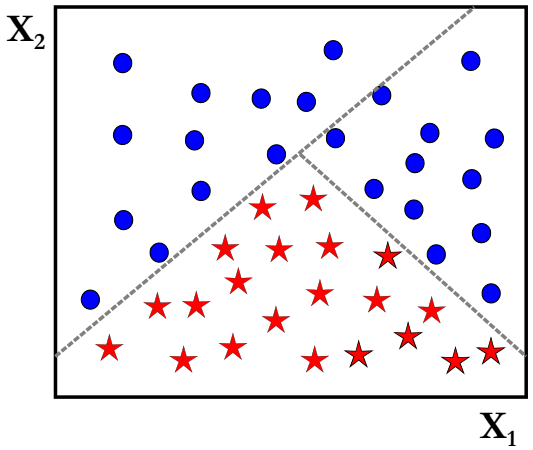
	X_1	X_2	X_3	X_4	X_5	X_6
1	x	x	x	x	x	x
2	x	x	x	x		
3	x	x	x	x		
4	x	x	x			
5	x	x				

Nonstandard data structure is not a problem for CART!

Important Characteristics of CART

- Capable of exploring *complex data sets* not easily handled by other techniques, where complexity can include:
  - High dimensionality
  - Mixture of data types
  - Nonstandard data structure
  - Nonhomogeneity

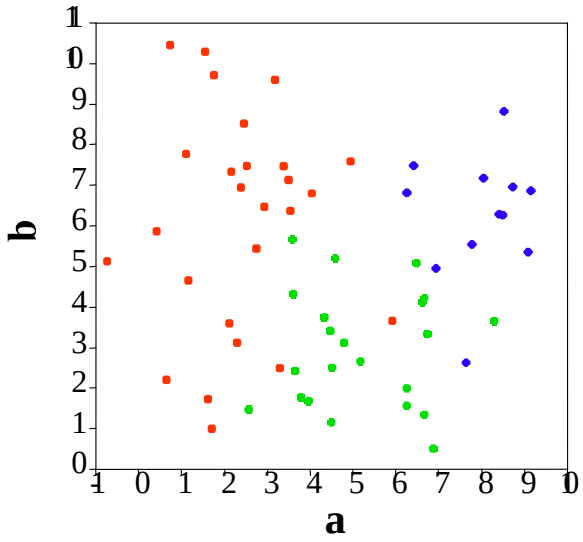
Different relationships hold between variables in different parts of the measurement space.



Nonhomogeneous data structure is not a problem for CART!

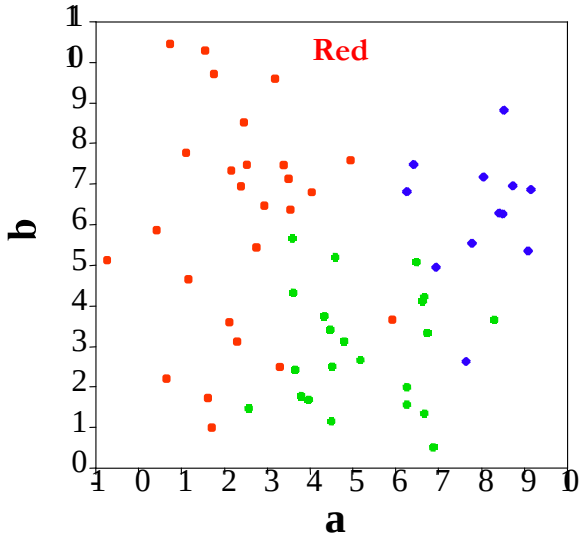
Classification and Regression Trees

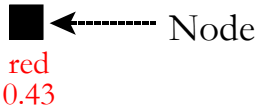
Growing Trees



Classification and Regression Trees

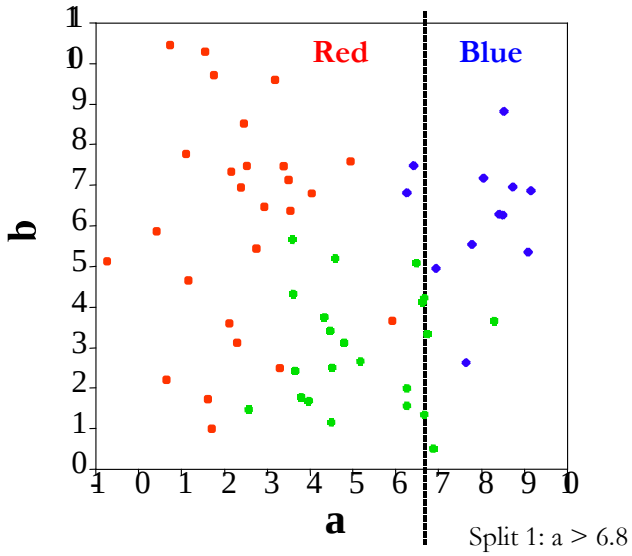
Growing Trees

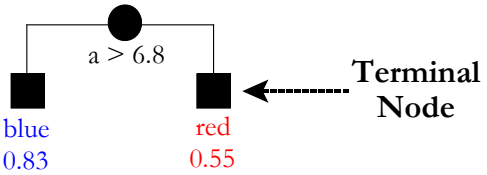




Classification and Regression Trees

Growing Trees

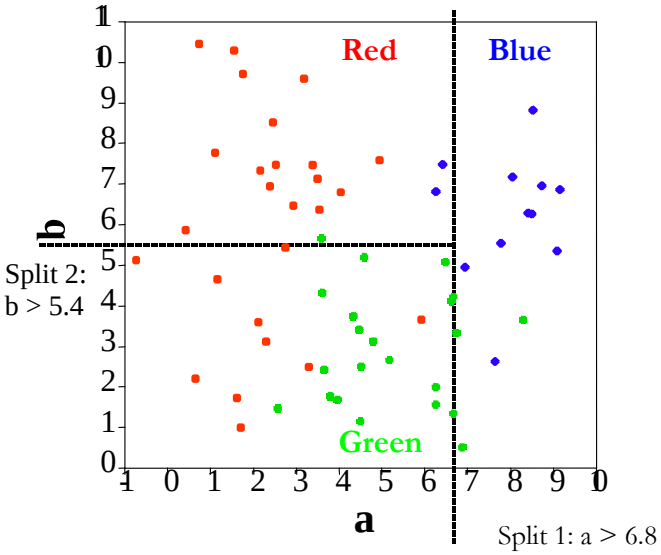


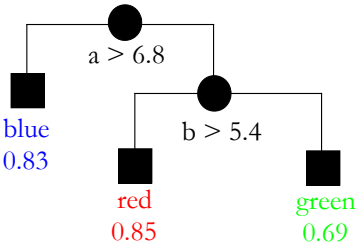


11

Classification and Regression Trees

Growing Trees

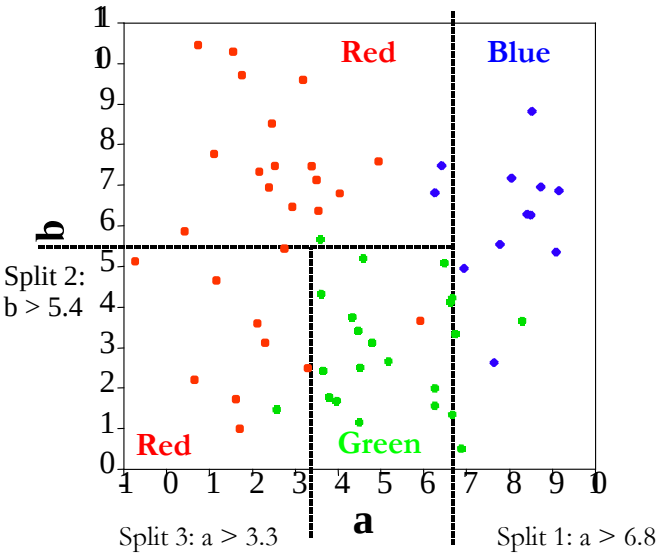


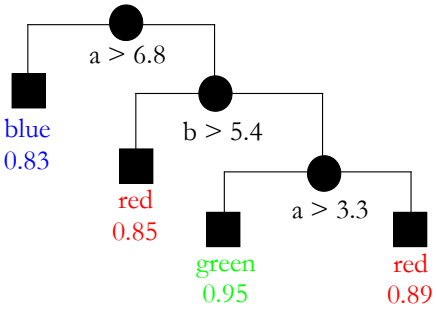


12

Classification and Regression Trees

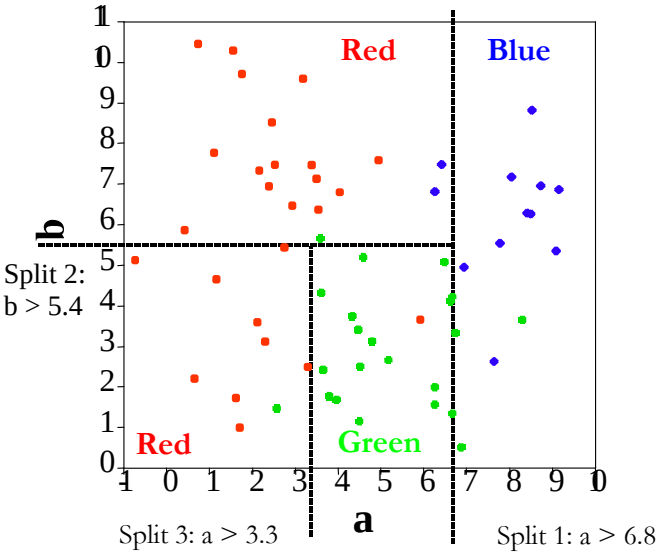
Growing Trees

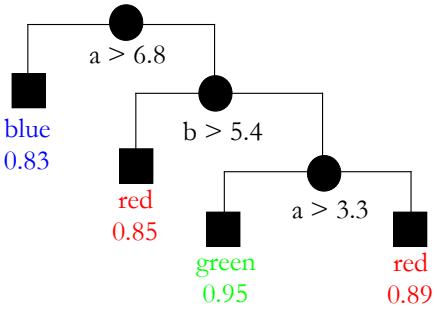




Classification and Regression Trees

Growing Trees



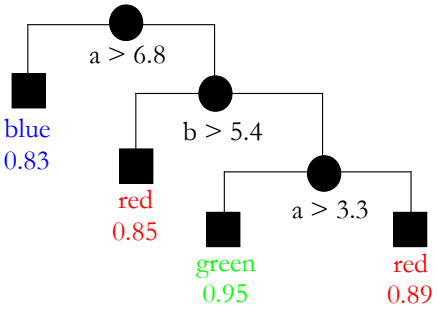


	red	green	blue
red	**25**	1	0
green	2	**18**	2
blue	2	0	**10**

Correct classification rate = 88%

Classification and Regression Trees
Growing Trees

1. At each node, the tree algorithm searches through the variables one by one, beginning with x_1 and continuing up to x_M.
2. For each variable, find best split.
3. Then it compares the M best single-variable splits and selects the best of the best.
4. Recursively partition each node until declared terminal.
5. Assign each terminal node to a class.



	red	green	blue
red	**25**	1	0
green	2	**18**	2
blue	2	0	**10**

Correct classification rate = 88%

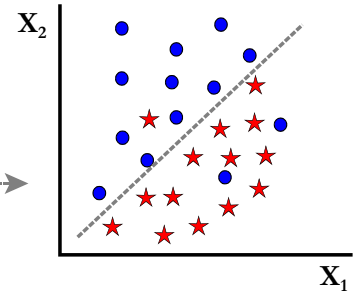
Growing Trees and Splitting Criteria
The Set of Questions, Q

- Each split depends on the value of only a *single* variable (but see below).
- For each *continuous* variable x_m, Q includes all questions of the form:
  - Is $x_m \le c_n$? ⟶ Is patch size ≤ 3 ha?
  - Where the c_n are taken halfway between consecutive distinct values of x_m
- If x_m is *categorical*, taking values $\{b_1, b_2, ..., b_L\}$, then Q includes all questions of the form:
  - Is $x_m \in s$? ⟶ Is cover type = A?
  - As s ranges over all subsets of $\{b_1, b_2, ..., b_L\}$

Growing Trees and Splitting Criteria

The Set of Questions, Q

- Q can be extended to include all *linear combination* splits of the form:
  - Is $\sum_m a_m x_m \leq c$?
- Q can be extended to include *Boolean combinations* of splits of the form:
  - Is ($x_m \in s$ and $x_m \in s$) or $x_m \in s$?



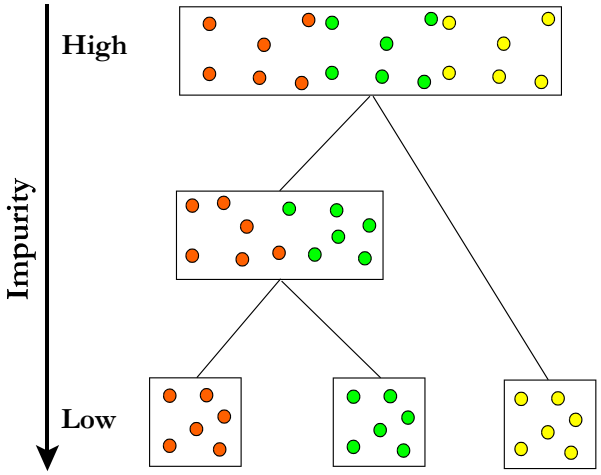
Does patient have (symptom A *and* symptom B) *or* symptom C?

Growing Trees and Splitting Criteria

Splitting Criteria

- The Concept of Node Impurity

- Node impurity refers to the mixing of classes among samples contained in the node.
  - Impurity is largest when all classes are equally mixed together.
  - Impurity is smallest when the node contains only one class.



Growing Trees and Splitting Criteria

Splitting Criteria

- Information Index:

$$i(t) = -\sum p(j/t) \ln p(j/t)$$

- Gini Index:

$$i(t) = 1 - \sum p^2(j/t)$$

$p(j/t)$ = probability that a case is in class j given that it falls into node t; equal to the proportion of cases at node t in class j if priors are equal to class sizes.

- Twoing Index:

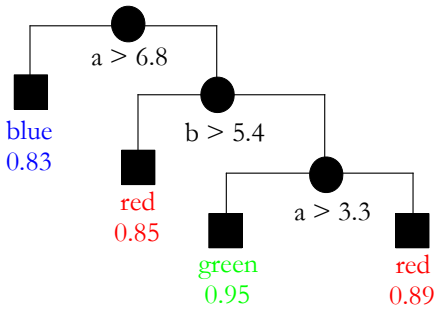
At every node, select the conglomeration of classes into two superclasses so that considered as a two-class problem, the greatest decrease in node impurity is realized.

Growing Trees and Splitting Criteria

Splitting Criteria

- At each node t, for each question in the set Q, the decrease in overall tree impurity is calculated as:

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R)$$



$i(t)$ = parent node impurity

p_L, p_R = proportion of observations answering 'yes' (descending to left node) and 'no' (descending to right node), respectively

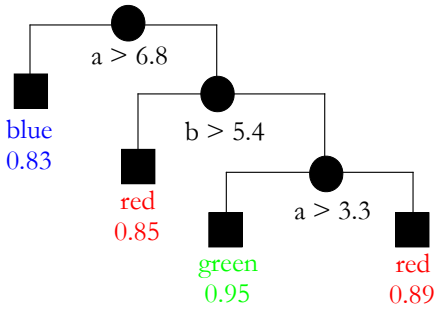
$i(t_{L,R})$ = descendent node impurity

Growing Trees and Splitting Criteria

Splitting Criteria

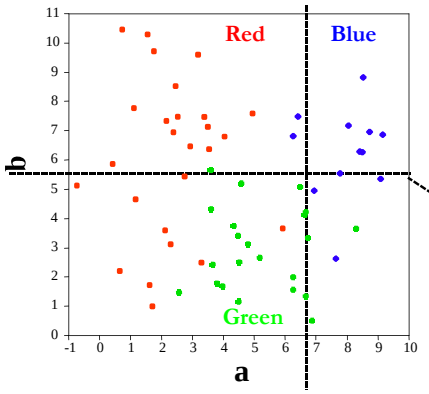
- At each node *t*, select the split from the set *Q* that maximizes the reduction in overall tree impurity.

$$Max\Delta i(s,t)$$



- Results are generally insensitive to the choice of splitting index, although the *Gini* index is generally preferred.
- Where they differ, the *Gini* tends to favor a split into one small, pure node and a large, impure node; *twoing* favors splits that tend to equalize populations in the two descendent nodes.

Growing Trees and Splitting Criteria

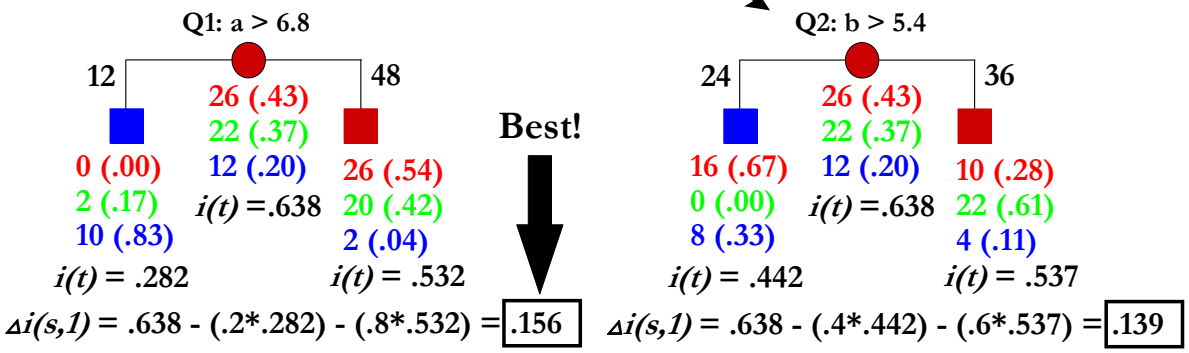


- Gini Index:

$$i(t) = 1 - \sum p^2(j/t)$$

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R)$$

$$Max\Delta i(s,t)$$

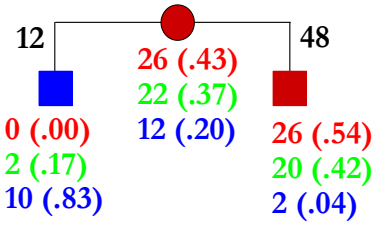


Growing Trees and Splitting Criteria

Class Assignment

- At each node *t*, assign the class containing the highest probability of membership in that node; i.e., the class that minimizes the probability of misclassification.

$$\max p(j/t) \longrightarrow$$ Equal to the proportion of cases in the largest class when priors are proportional.



R(T) **= (.17*.2)+(.46*.8) = .39**

Resubstitution estimate of the probability of misclassification, given that a case falls into node *t*:

$$r(t) = 1 - \max p(j/t)$$

Overall tree misclassification rate:

$$R(T) = \sum_{t \in T} r(t)\, p(t)$$

Growing Trees and Splitting Criteria

The Role of Prior Probabilities

- Prior probabilities affect both the split selected and the class assignment made at each node.

Prior probability that a class *j* case will be presented to the tree: $\longrightarrow \pi(j)$

Resubstitution estimate of the probability that a case will be in both class *j* and fall into node *t*: $\longrightarrow p(j,t) = \pi(j) \cdot N_j(t) / N_j$

Probability that a case is in class *j* given that it falls into node *t*: $\longrightarrow p(j/t) = p(j,t) \Big/ \sum_j p(j,t)$

Growing Trees and Splitting Criteria

The Role of Prior Probabilities

- Gini Splitting Index: $i(t) = 1 - \sum p^2(j/t)$

When priors match class proportions in sample: $\pi(j) = N_j / N$

Resubstitution estimate of the probability that a case will be in both class j and fall into node t: $p(j,t) = N_j(t) / N$

Probability that a case is in class j given that it falls into node t: $p(j/t) = N_j(t) / N(t)$

Relative proportions of class *j* cases in node *t*

Growing Trees and Splitting Criteria

The Role of Prior Probabilities

- Gini Splitting Index: $i(t) = 1 - \sum p^2(j/t)$

When priors are user specified: $\pi(j)$

Resubstitution estimate of the probability that a case will be in both class j and fall into node t: $p(j,t) = \pi(j) \cdot N_j(t) / N_j$

Probability that a case is in class j given that it falls into node t: $p(j/t) = p(j,t) / \sum_j p(j,t)$

Growing Trees and Splitting Criteria

The Role of Prior Probabilities

- Class Assignment: $\max p(j/t)$

When priors match class proportions in sample: $\pi(j) = N_j / N$

Resubstitution estimate of the probability that a case will be in both class j and fall into node t: $p(j,t) = N_j(t) / N$

Probability that a case is in class j given that it falls into node t: $p(j/t) = N_j(t) / N(t)$

Relative proportions of class *j* cases in node *t*

Growing Trees and Splitting Criteria

The Role of Prior Probabilities

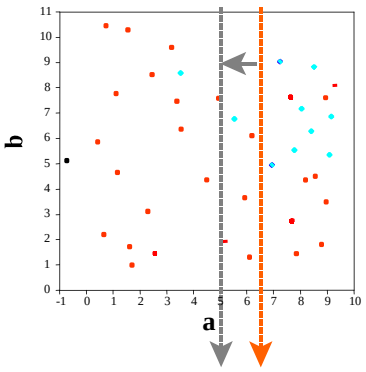
- Class Assignment: $\max p(j/t)$

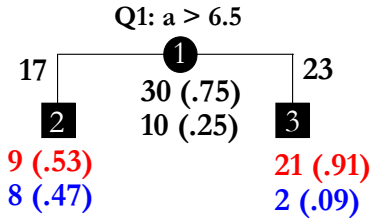
When priors are user specified: $\pi(j)$

Resubstitution estimate of the probability that a case will be in both class j and fall into node t: $p(j,t) = \pi(j) \cdot N_j(t) / N_j$

Probability that a case is in class j given that it falls into node t: $p(j/t) = p(j,t) / \sum_j p(j,t)$

Growing Trees and Splitting Criteria

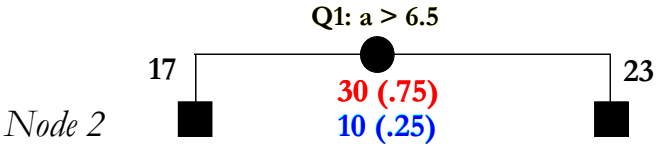




Adjusting Prior Probabilities

	Red	Blue	Red	Blue	Red	Blue
Priors:	.75	.25	.50	.50	.20	.80
Node 1:						
#	30	10	30	10	30	10
p(j/1)	.75	.25	.50	.50	.20	.80
Gini	.375		.50		.32	
Node 2:						
#	9	8	9	8	9	8
p(j/2)	.529	.471	.272	.727	.086	.914
Gini	.498		.397		.157	
Class	Red		Blue		Blue	
Node 3:						
#	21	2	21	2	21	2
p(j/3)	.913	.086	.778	.222	.467	.533
Gini	.159		.346		.498	
Class	Red		Red		Blue	

Growing Trees and Splitting Criteria



Priors Proportional

9 [*p(j,t)* = .75*.3 = .225]
[*p(j/t)* = .225/.425 = .53]

8 [*p(j,t)* = .25*.8 = .2]
[*p(j/t)* = .2/.425 = .47]

max *p(j/t)* = Red

21 [*p(j,t)* = .75*.7 = .525]
[*p(j/t)* = .525/.53 = .91]

2 [*p(j,t)* = .25*.2= .05]
[*p(j/t)* = .05/.53 = .09]

max *p(j/t)* = Red

Priors Equal

9 [*p(j,t)* = .5*.3 = .15]
[*p(j/t)* = .15/.55 = .27]

8 [*p(j,t)* = .5*.8 = .4]
[*p(j/t)* = .4/.55 = .73]

max *p(j/t)* = Blue

21 [*p(j,t)* = .5*.7 = .35]
[*p(j/t)* = .35/.45 = .78]

2 [*p(j,t)* = .5*.2= .1]
[*p(j/t)* = .1/.45 = .22]

max *p(j/t)* = Red

Growing Trees and Splitting Criteria

Variable Missclassification Costs

- Variable missclassification costs can be specified, which can affect both the split selected and the class assignment made at each node.

		i = 1	2	3	4
j =	1	0	5	5	5
	2	1	0	1	1
	3	1	1	0	1
	4	1	1	1	0

$c(i|j) \geq 0, \quad i \neq j$

$c(i|j) = 0, \quad i = j$

Where $c(i|j)$ = cost of missclassifying a class j object into class i

Five times more costly to missclassify a case from class 1!

Growing Trees and Splitting Criteria

Variable Missclassification Costs

- Variable missclassification costs can be directly incorporated into the splitting rule.

- Gini Splitting Index:

$$i(t) = 1 - \sum_j p^2(j/t) \quad \text{or} \quad i(t) = \sum_{j \neq i} p(i/t)\,p(j/t)$$

- Symmetric Gini Index:

$$i(t) = \sum_{j,i} c(i/j)\,p(i/t)\,p(j/t)$$

Where $c(i|j)$ = cost of missclassifying a class j object into class i

Growing Trees and Splitting Criteria

Variable Missclassification Costs

- Variable missclassification costs can be accounted for by adjusting the prior probabilities when costs are constant for class *j*.

- Altered Priors:

j \	1	2	3
1	0	5	5
2	1	0	1
3	1	1	0

$$\pi'(j) = c(j)\pi(j) \Big/ \sum_j c(j)\pi(j)$$

Preferred under constant, asymmetric cost structure

- Symmetric Gini Index:

j \	1	2	3
1	0	3	3
2	3	0	1
3	3	1	0

$$i(t) = \sum_{j,i} c(i/j)\,p(i/t)\,p(j/t)$$

Preferred under symmetric cost structure

Growing Trees and Splitting Criteria

Variable Missclassification Costs

- At each node *t*, assign the class that minimizes the expected missclassification cost (instead of missclassification probability).

$$\sum_j c(i/j)\,p(j/t) \longrightarrow$$

Where *c(i|j)* = cost of missclassifying a class *j* object into class *i*

Resubstitution estimate of the expected missclassification cost, given the node *t*:

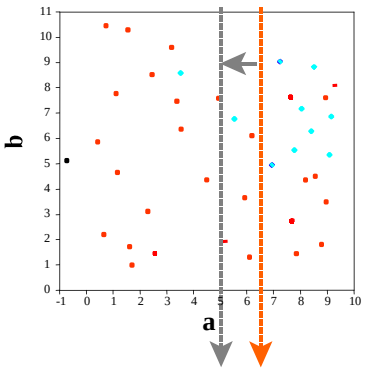
$$r(t) = \min_i \sum_j c(i/j)\,p(j/t)$$

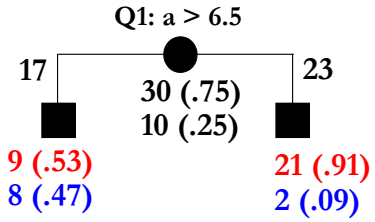
Overall tree missclassification cost:

$$R(t) = \sum_{t \in T} r(t)\,p(t)$$

Growing Trees and Splitting Criteria

Adjusting Missclassification Costs





	Red	Blue	Red	Blue	Red	Blue
Priors:	.75	.25	.75	.25	.75	.25
Costs:	1	1	1	2	1	12
Node 1:						
#	30	10	30	10	30	10
cp(j/1)	.25	.75	.50	.75	3.00	.75
S-Gini	.375		.563		2.438	
Node 2:						
#	9	8	9	8	9	8
cp(j/2)	.471	.529	.941	.529	5.647	.529
S-Gini	.498		.747		3.239	
Class	Red		Blue		Blue	
Node 3:						
#	21	2	21	2	21	2
cp(j/3)	.086	.913	.174	.913	1.043	.913
S-Gini	.159		.238		1.032	
Class	Red		Red		Blue	

Growing Trees and Splitting Criteria

Key Points Regarding Priors and Costs

- Priors represent the probability of a class *j* case being submitted to the tree.
  - Increasing prior probability of class *j* increases the relative importance of class *j* in defining the best split at each node and increases the probability of class assignment at terminal nodes (i.e., reduces class *j* missclassifications.
    - Priors proportional = max overall tree correct classification rate
    - Priors equal = max(?) mean class correct classification rate (strives to achieve better balance among classes)

Growing Trees and Splitting Criteria

Key Points Regarding Priors and Costs

- Costs represent the 'cost' of missclassifying a class *j* case.
  - Increasing cost of class *j* increases the relative importance of class *j* in defining the best split at each node and increases the probability of class assignment at terminal nodes (i.e., reduces class *j* missclassifications).
    - Asymmetric but constant costs = can use Gini Index with altered priors (but see below)
    - Symmetric and nonconstant = use Symmetric Gini Index (i.e., incorporate costs directly)

Growing Trees and Splitting Criteria

Key Points Regarding Priors and Costs

There are two ways to adjust splitting decisions and final class assignments to favor one class over another.

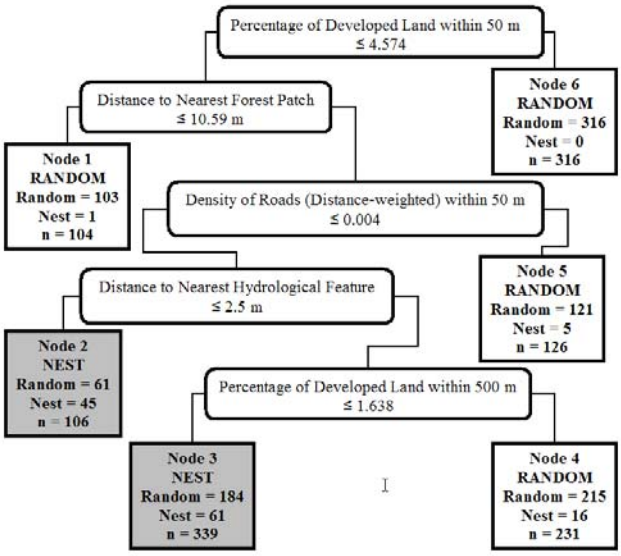
Should I alter Priors or adjust Costs?

- My preference is to adjust *priors* to reflect *inherent* property of the population being sampled (if relative sample sizes don't represent priors, then null approach is to assume equal priors).
- Adjust *costs* to reflect *decisions* regarding the intended use of the classification tree.

Growing Trees and Splitting Criteria

Harrier Example of Setting Priors and Costs

"Preferred" Nesting Habitat



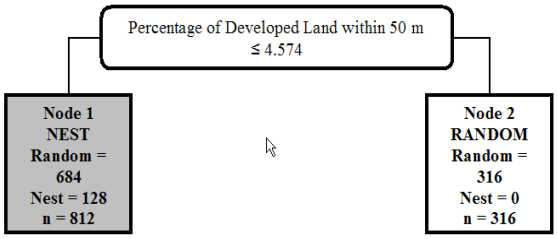
- 128 nests; 1000 random
- equal priors
- *equal costs*

39

Growing Trees and Splitting Criteria

Harrier Example of Setting Priors and Costs

"Potential" Nesting Habitat



Characteristic	Classification Tree	
	Potential	Preferred
Misclassification cost ratio (nest:random)	3:1	1:1
Correct classification rate of nests (%)	100	83
Total area (ha)	7355	2649
Area protected (ha)	5314	2266
Area protected (%)	72	86

- 128 nests; 1000 random
- equal priors
- *3:1 costs*

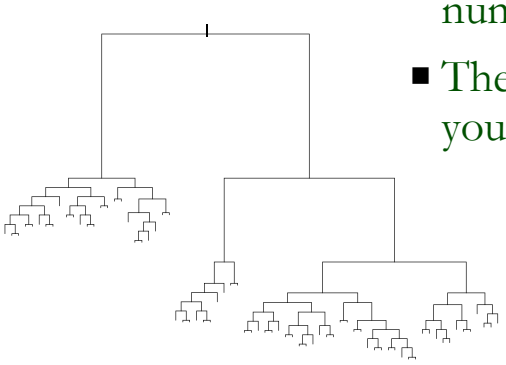
40

Selecting the Right-Sized Trees

- The most significant issue in CART is not growing the tree, but deciding when to stop growing the tree or, alternatively, deciding on the right size tree.

The crux of the problem:

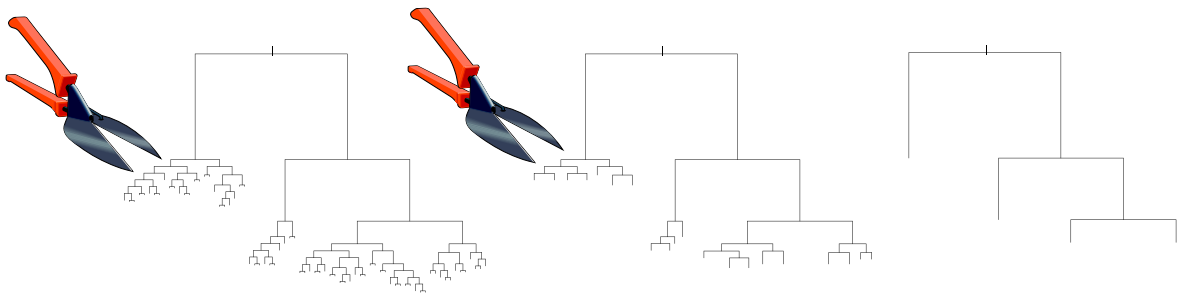
- Missclassification rate decreases as the number of terminal nodes increases.
- The more you split, the better you think you are doing.
- Yet, overgrown trees do not produce *"honest"* missclassification rates when applied to new data.



Selecting the Right-Sized Trees

Minimum Cost-Complexity Pruning

- The solution is to overgrow the tree and then prune it back to a smaller tree that has the minimum honest estimate of true (prediction) error.
  - The preferred method is based on *minimum cost-complexity pruning* in combination with *V-fold cross-validation*.



Selecting the Right-Sized Trees

Minimum Cost-Complexity Pruning

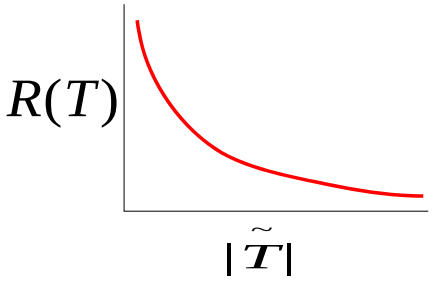
Cost-complexity measure:

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|$$

$R(T)$ = Overall missclassification cost

α = Complexity parameter (complexity cost per terminal node)

$|\tilde{T}|$ = Complexity = # terminal nodes

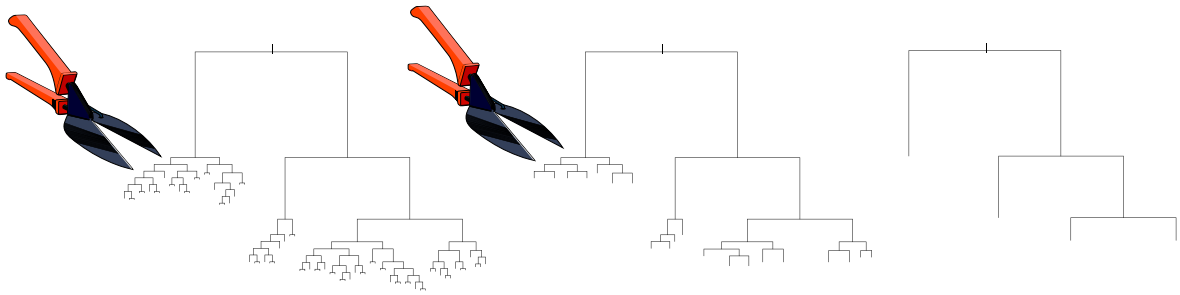


Minimum cost-complexity pruning recursively prunes off the weakest-link branch, creating a nested sequence of subtrees.

Selecting the Right-Sized Trees

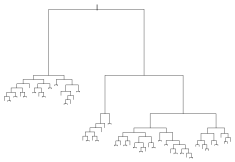
Pruning with Cross-Validation

- But we need an unbiased (honest) estimate of missclassification costs for trees of a given size.
  - Preferred method is to use *V-fold cross-validation* to obtain honest estimates of true (prediction) error for trees of a given size.



Selecting the Right-Sized Trees

Pruning with Cross-Validation



V-fold Cross-Validation:

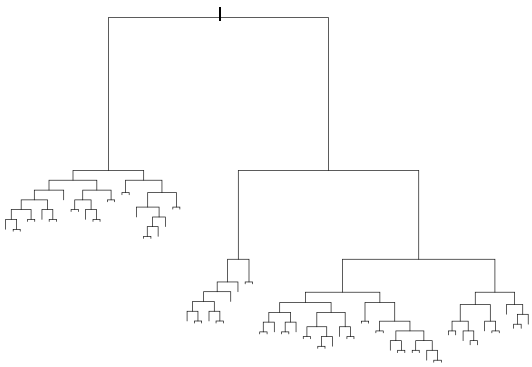
1. Divide the data into V mutually exclusive subsets of approximately equal size.
2. Drop out each subset in turn, build a tree using data from the remaining subsets, and use it to predict the response for the omitted subset.
3. Calculate the estimated error for each subset, and sum over all subsets.
4. Repeat steps 2-3 for each size of tree.
5. Select the tree with the minimum estimated error rate (but see below).

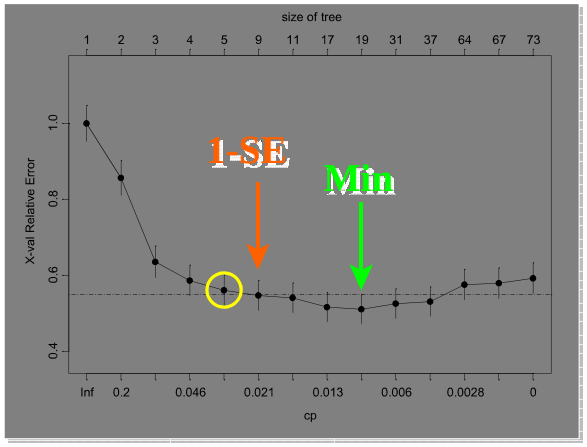
Selecting the Right-Sized Trees

Pruning with Cross-Validation

1-SE Rule

- The 'best' tree is taken as the smallest tree such that its estimated error is within one standard error of the minimum.

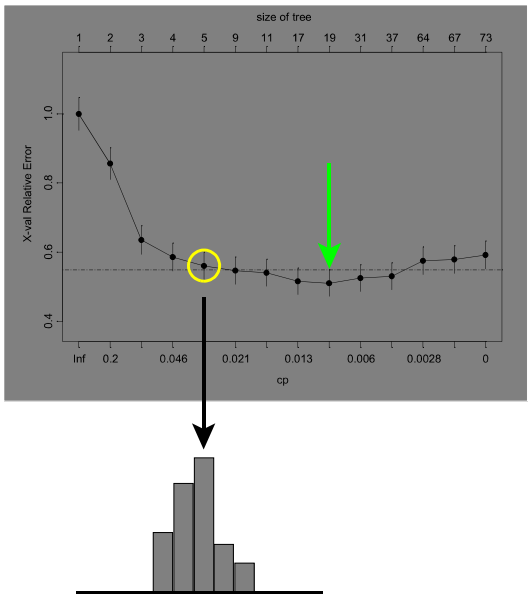




Selecting the Right-Sized Trees

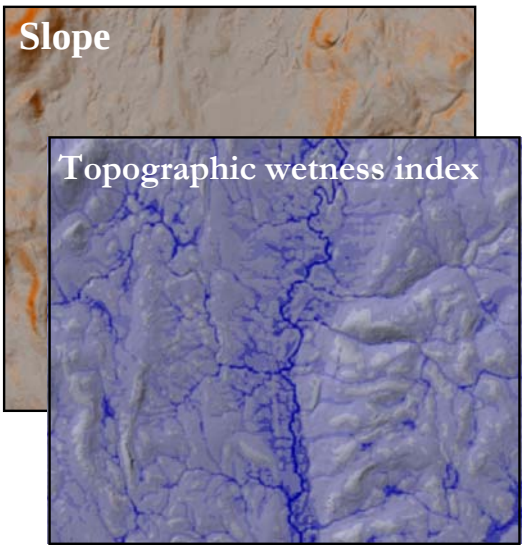
Repeated Cross-Validation

- Repeat the V-fold cross-validation many times and select the tree size from each cross-validation of the series according to the 1-SE rule.
- From the distribution of selected tree sizes, select the most frequently occurring (modal) tree size.

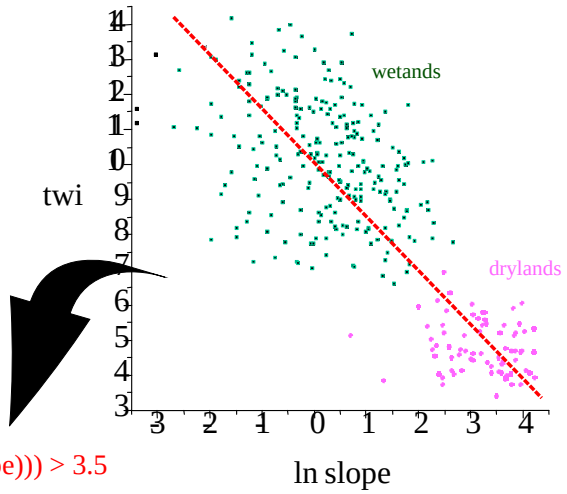


Classification and Regression Trees

Illustrated Example



- Communities split a priori into 3 groups
  - Wetland only
  - Dryland only
  - May be in either (mediumlands)



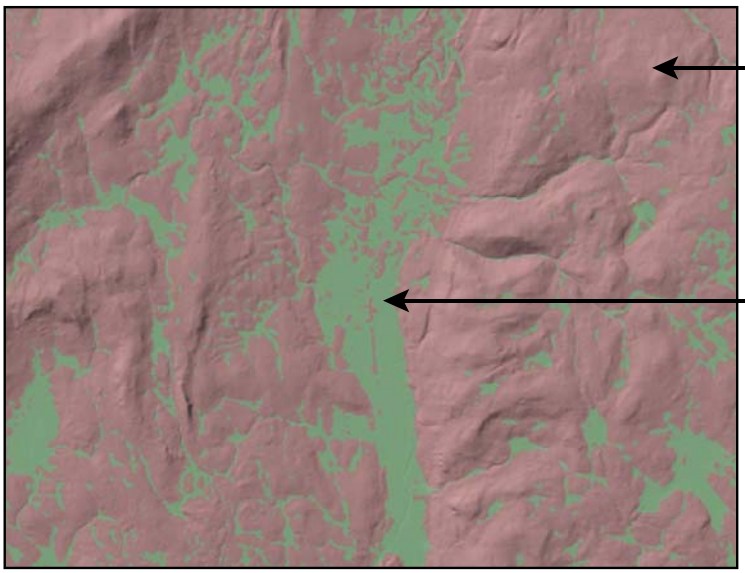
Discriminant Function

wet = ((.4725635 * twic2) + (-0.4528466 * ln (slope))) > 3.5

Classification and Regression Trees

Illustrated Example

Wetlands vs. Drylands



Areas that can have drylands or mediumlands

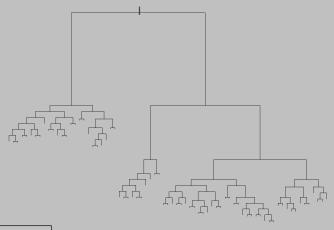
Areas that can have wetlands or mediumlands

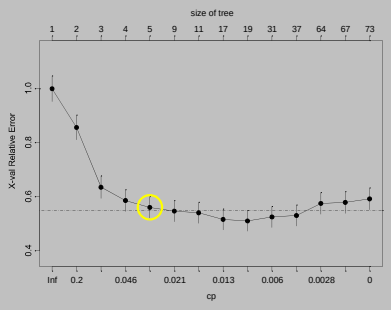
All further classification is conditioned on this partition

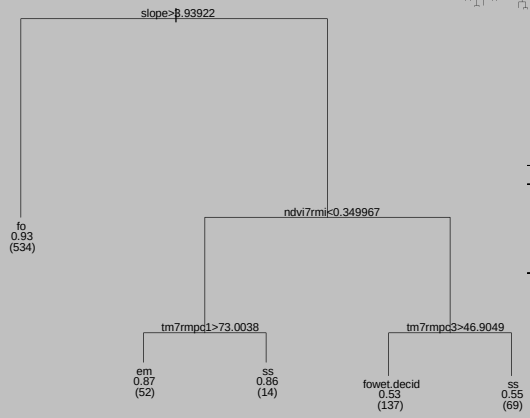
An example: classifying wetlands with CART

Wetland data

fo	519
fowet.decid	119
fowet.conif	27
ss	75
em	66
other	429







Misclassification rates: Null = 36%, Model = 18% (143/806)

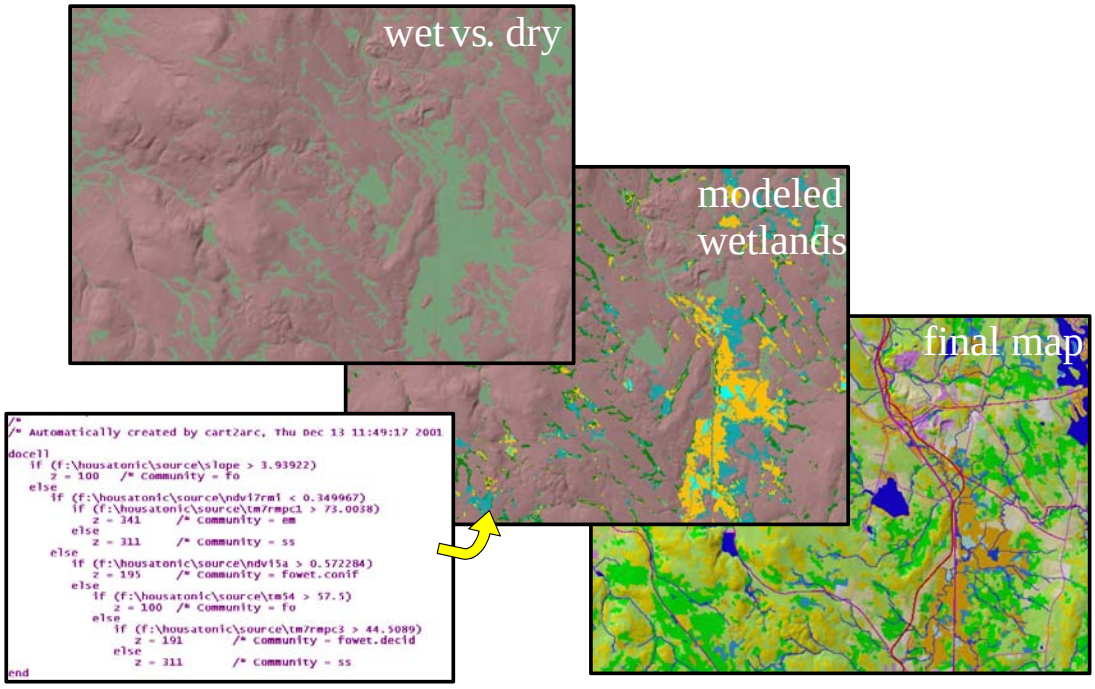
Confusion matrix

	fo	fowet.decid	fowet.conif	ss	em
fo	495	21	3	7	8
fowet.decid	21	73	24	13	6
fowet.conif	0	0	0	0	0
ss	3	23	0	50	7
em	0	2	0	5	45

Correct classification rate = 82%

Classification and Regression Trees

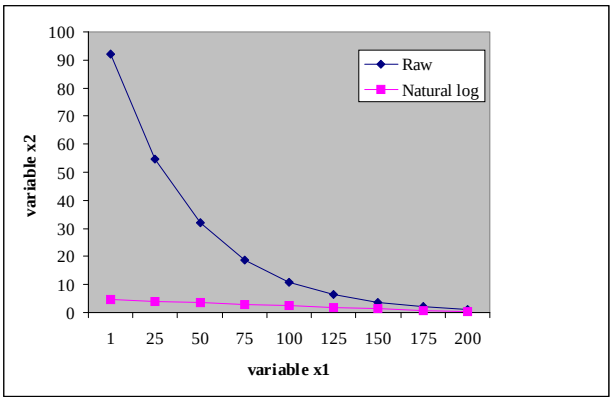
Illustrated Example



Other CART Issues

Transformations

- For a numeric explanatory variable (continuous or count), only its rank order determines a split. Trees are thus invariant to monotonic transformations of such variables.



Other CART Issues

Surrogate Splits and Missing Data

- Cases with missing explanatory variables can be handled through the use of surrogate splits.

1. Define a measure of similarity between any two splits, s, s' of a node t.
2. If the best split of t is the split s on the variable x_m, find the split s' on the variables other than x_m that is most similar to s.
3. Call s' the best surrogate for s.
4. Similarly, define the second best surrogate split, third best, and so on.
5. If a case has x_m missing, decide whether it goes to t_L or t_R by using the best available surrogate split.

Other CART Issues

Surrogate Splits and Variable Importance

Variable Ranking:

- How to assess the overall importance of a variable, despite whether or not it occurs in any split in the final tree structure?

➡ A variable can be '*masked*' by another variable that is slightly better at each split.

The measure of importance of variable x_m is defined as:

$$M(x_m) = \sum_{t \in T} \Delta I(\tilde{s}_m, t)$$

Where: $\tilde{s}_m$ = surrogate split on x_m at node t

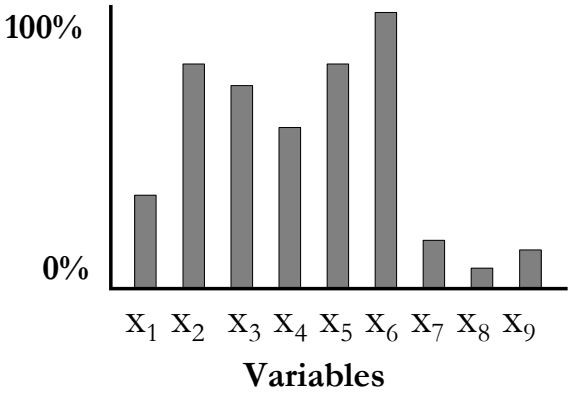
Other CART Issues

Surrogate Splits and Variable Importance

Variable Ranking:

- Since only the relative magnitude of the $M(x_m)$ are interesting, the actual measures of importance are generally given as the normalized quantiles:

$$100\,M(x_m) \Big/ \max_m M(x_m)$$



***Caution**, the size of the tree and the number of surrogate splitters considered will affect variable importance!

Other CART Issues

Competing Splits and Alternative Structures

- At any given node, there may be a number of splits on different variables, all of which give almost the same decrease in impurity. Since data are noisy, the choice between *competing splits* is almost random. However, choosing an alternative split that is almost as good will lead to a different evolution of the tree from that node downward.
  - For each split, we can compare the strength of the split due to the selected variable with the best splits of each of the remaining variables.
  - A strongly competing alternative variable can be substituted for the original variable, and this can sometimes simplify a tree by reducing the number of explanatory variables or lead to a better tree.

Other CART Issues

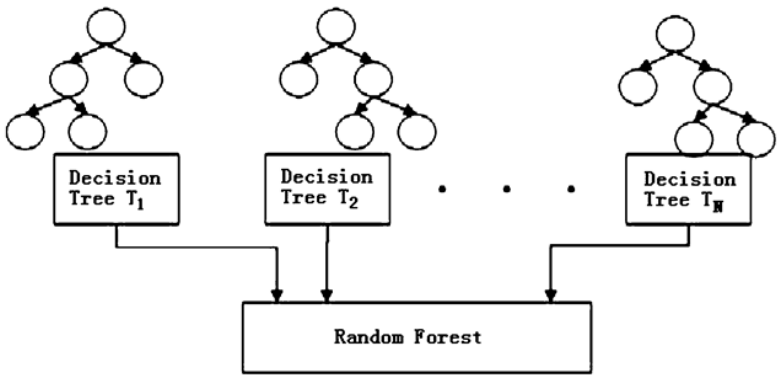
Random Forests (Bagged Trees)

- A *random forest* consists of many trees all based on the same model (i.e., set of variables, splitting rule, etc.).
- Each tree in the forest is grown using a *bootstrapped* version of a learn sample ($2/3^{rds}$ of observations) and a random subset of the variables (usually sqrt(p)) at each node, so that each tree in the committee is somewhat different from the others.
- Each tree is grown out fully; i.e., there is *no pruning* involved since it can be shown that overfitting is not a problem.
- Unbiased classification error is estimated by submitting the "*oob*" (out-of-bag) hold-out observations to each tree and averaging error rates across all the trees in the random forest.

Other CART Issues

Random Forests (Bagged Trees)

- The random forest (sometimes also referred to a "committee of scientists") generates predictions on a case-by-case basis by "*voting*". Each tree in the forest produces its own predicted response. The ultimate prediction is decided by majority rule.



Other CART Issues

Random Forests (Bagged Trees)

- Variable Importance:

1) *Mean decrease in accuracy* = For each tree, the prediction error on the oob portion of the data is recorded. Then the same is done after permuting each predictor variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences.

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
openwater	18.91	17.42	21.02	4.04
depth	4.39	6.20	6.86	1.20
canopy	5.85	2.81	5.55	1.21
forest	17.24	17.46	19.55	3.99
VP	2.42	3.29	3.34	0.39
razz	6.89	10.25	10.57	1.45
shroom	5.84	4.65	6.92	1.20
herb	0.90	-0.15	0.44	0.54
slug	13.08	12.82	16.05	3.50
worm	7.37	8.63	10.35	1.81
em	-0.30	1.14	0.70	0.45
fowet	1.60	2.23	2.56	0.19
riv	0.85	1.10	1.45	0.24
ss	-0.09	0.94	0.58	0.67
ub	6.68	6.00	7.40	0.56
upland	1.30	1.34	1.34	0.49
edgefine	2.22	1.53	2.42	0.55
edgecoarse	1.63	-0.95	0.32	0.45
Pafine	2.78	2.83	3.45	0.36
Pacoarse	2.27	0.40	2.35	0.27
strlen	12.60	9.50	13.30	2.53
distlotic	-0.62	2.80	1.44	0.62
distH2O	0.22	2.32	1.76	0.53
distfowet	0.28	4.30	3.00	0.86
distupland	8.73	9.02	10.63	1.95

Other CART Issues

Random Forests (Bagged Trees)

- Variable Importance:

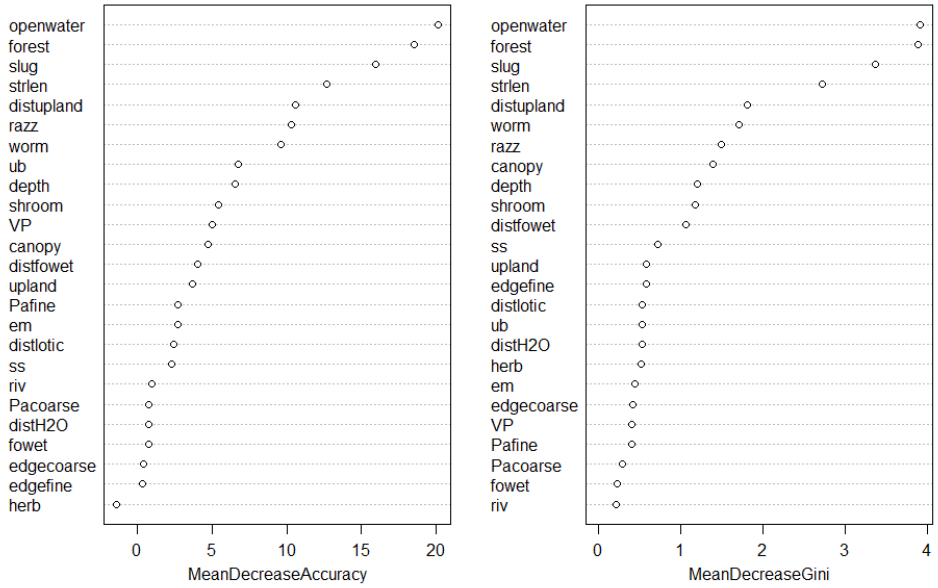
2) *Mean decrease in Gini* = total decrease in node impurities from splitting on the variable, averaged over all trees.

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
openwater	18.91	17.42	21.02	4.04
depth	4.39	6.20	6.86	1.20
canopy	5.85	2.81	5.55	1.21
forest	17.24	17.46	19.55	3.99
VP	2.42	3.29	3.34	0.39
razz	6.89	10.25	10.57	1.45
shroom	5.84	4.65	6.92	1.20
herb	0.90	-0.15	0.44	0.54
slug	13.08	12.82	16.05	3.50
worm	7.37	8.63	10.35	1.81
em	-0.30	1.14	0.70	0.45
fowet	1.60	2.23	2.56	0.19
riv	0.85	1.10	1.45	0.24
ss	-0.09	0.94	0.58	0.67
ub	6.68	6.00	7.40	0.56
upland	1.30	1.34	1.34	0.49
edgefine	2.22	1.53	2.42	0.55
edgecoarse	1.63	-0.95	0.32	0.45
Pafine	2.78	2.83	3.45	0.36
Pacoarse	2.27	0.40	2.35	0.27
strlen	12.60	9.50	13.30	2.53
distlotic	-0.62	2.80	1.44	0.62
distH2O	0.22	2.32	1.76	0.53
distfowet	0.28	4.30	3.00	0.86
distupland	8.73	9.02	10.63	1.95

Other CART Issues

Random Forests (Bagged Trees)

- Variable Importance:

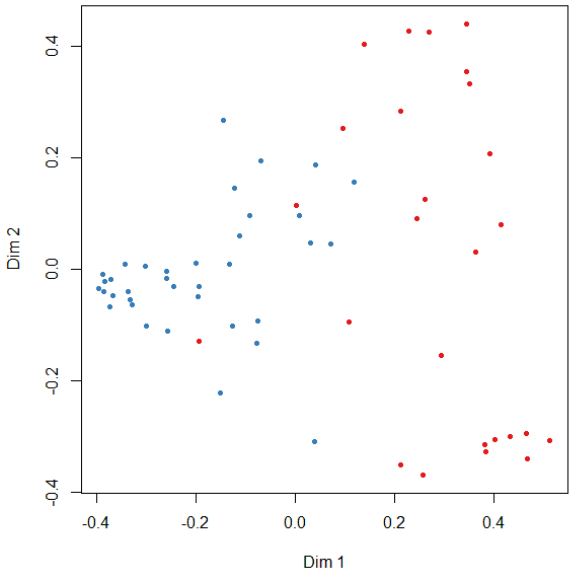


Other CART Issues

Random Forests (Bagged Trees)

- Proximity:

Measure of proximity (or similarity) based on how often any two (oob) observations end up in the same terminal node. 1-prox(k,n) form Euclidean distances in a high dimensional space that can be projected down onto a low dimensional space using metric scaling (aka principal coordinates analysis) to give an informative view of the data.



Other CART Issues

Random Forests (Bagged Trees)

- Variable Importance:

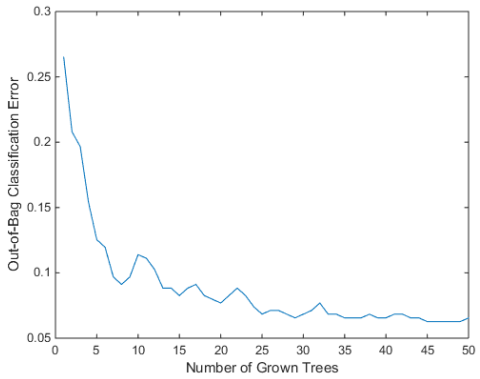
2) *Mean decrease in Gini* = total decrease in node impurities from splitting on the variable, averaged over all trees.

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
openwater	18.91	17.42	21.02	4.04
depth	4.39	6.20	6.86	1.20
canopy	5.85	2.81	5.55	1.21
forest	17.24	17.46	19.55	3.99
VP	2.42	3.29	3.34	0.39
razz	6.89	10.25	10.57	1.45
shroom	5.84	4.65	6.92	1.20
herb	0.90	-0.15	0.44	0.54
slug	13.08	12.82	16.05	3.50
worm	7.37	8.63	10.35	1.81
em	-0.30	1.14	0.70	0.45
fowet	1.60	2.23	2.56	0.19
riv	0.85	1.10	1.45	0.24
ss	-0.09	0.94	0.58	0.67
ub	6.68	6.00	7.40	0.56
upland	1.30	1.34	1.34	0.49
edgefine	2.22	1.53	2.42	0.55
edgecoarse	1.63	-0.95	0.32	0.45
Pafine	2.78	2.83	3.45	0.36
Pacoarse	2.27	0.40	2.35	0.27
strlen	12.60	9.50	13.30	2.53
distlotic	-0.62	2.80	1.44	0.62
distH2O	0.22	2.32	1.76	0.53
distfowet	0.28	4.30	3.00	0.86
distupland	8.73	9.02	10.63	1.95

Other CART Issues

Boosted Classification & Regression Trees

- **Boosting** involves growing a sequence of trees, with successive trees grown on reweighted versions of the data. At each stage of the sequence, each data case is classified from the current sequence of trees, and these classifications are used as weights for fitting the next tree of the sequence. Incorrectly classified cases receive more weight than those that are correctly classified, and thus cases that are difficult to classify receive ever-increasing weight, thereby increasing their chance of being correctly classified. The final classification of each case is determined by the weighted majority of classifications across the sequence of trees.



Limitations of CART

- CART's *greedy algorithm* approach in which all possible subsets of the data are evaluated to find the "best" split at each node has two problems:
  - Computational Complexity...the number of possible splits and subsequent node impurity calculations can be computionally challenging.
  - Bias in Variable Selection...
    - Ordered variable: #splits = n-1 n = #distinct values
    - Categorical variable: #splits = $2^{(m-1)} - 1$ m = #classes
    - Linear combinations of ordered variables: $\sum_{k=1}^{K} a_k x_k \le c$ #splits = $\{a_1,...a_K,c\}$

Limitations of CART

- CART's greedy algorithm approach in which all possible subsets of the data are evaluated to find the "best" split at each node has two problems:
  - Computational Complexity...
  - Bias in Variable Selection...unrestricted search tends to select variables that have more splits, making it hard to draw reliable conclusions from the tree structures
    - Alternative approaches exist (QUEST) in which variable selection and split point selection are done separately. At each node, an ANOVA F-statistic is calculated (after a DA transformation of categorical variables) and the variable with the largest F-statistic is selected, and then quadratic two-group DA is applied to find the best split (if >2 groups, then 2-group NHC is used to create 2 superclasses).

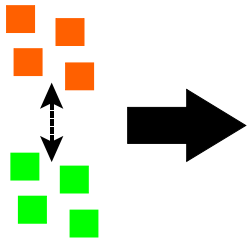
Caveats on CART

- CART is a *nonparametric* procedure, meaning that it is not based on an underlying theoretical model, in contrast to Discriminant Analysis, which is based on a linear response model, and Logistic Regression which is based on a logistic (logit) response model.

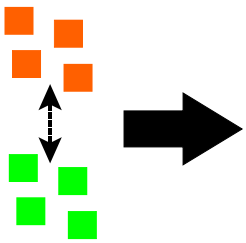
What are the implications of this?

- Lack of theoretical underpinning precludes an explicit test of ecological theory?
- Yet the use of an empirical model means that our inspection of the data structure is not constrained by the assumptions of a particular theoretical model?

Discrimination Among Groups



- Are groups significantly different? (How valid are the groups?)
  - Multivariate Analysis of Variance (MANOVA)
  - Multi-Response Permutation Procedures (MRPP)
  - Analysis of Group Similarities (ANOSIM)
  - Mantel's Test (MANTEL)

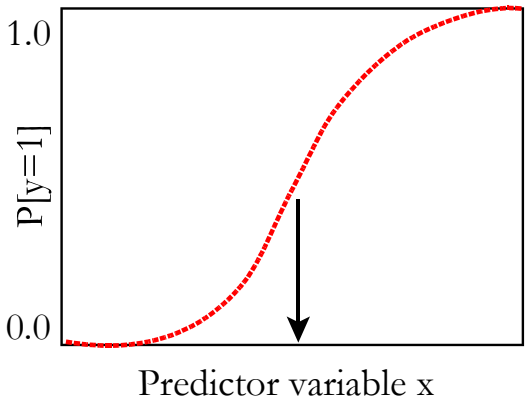


- How do groups differ? (Which variables best distinguish among the groups?)
  - Discriminant Analysis (DA)
  - Classification and Regression Trees (CART)
  - Logistic Regression (LR)
  - Indicator Species Analysis (ISA)

Logistic Regression

- *Parametric* procedure useful for exploration, description, and prediction of grouped data (Hosmer and Lemenshow 2000).

- In the simplest case of *2 groups* and a single predictor variable, LR predicts group membership as a *sigmoidal* prbabilistic function for which the prediction 'switches' from one group to the other at a critical value, typically at the inflection point or value where P[y=1] = 0.5, but it can be specified otherwise.



Logistic Regression

- Logistic Regression can be generalized to include several predictor variables and multiple (multinomial) categorical states or groups.

- Exponent term is a regression equation that specifies a *logit* (log of the odds) model which is a likelihood ratio that contrasts the probability of membership in one group to that of another group.

$$p[k] = \frac{e^{(b_0 + b_1 x_{k1} + b_2 x_{k2} + \ldots + b_m x_{km})}}{1 + e^{(b_0 + b_1 x_{k1} + b_2 x_{k2} + \ldots + b_m x_{km})}}$$

$$\ln\left(\frac{p[k]}{1 - p[k]}\right) = b_0 + b_1 x_{k1} + b_2 x_{k2} + \ldots + b_m x_{km}$$

Logistic Regression

- Can be applied to any data structure, including mixed data sets containing both continuous, categorical, and count variables, but requires standard data structures.
- Like DA, cannot handle missing data.
- Final classification has a simple form which can be compactly stored and that efficiently classifies new data.
- Can be approached from a model-selection standpoint, using goodness-of-fit tests or cost complexity measures (AIC) to choose the simplest model that fits the data.
- Regression coefficients offer the same interpretive aid as in other regression-based analyses.

Indicator Species Analysis

- *Nonparametric* procedure for distinguishing among groups based on species compositional data, where the goal is to identify those species that show high fidelity to a particular group and as such can serve as indicators for that group (Dufrene and Legendre 1997).

- Compute the mean within-group abundance of each species *j* in each group *k*.

$$\bar{x}_{jk} = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ijk}$$

- Compute an index of *relative abundance* within each group.

$$RA_{jk} = \bar{x}_{jk} \Big/ \sum_{k=1}^{g} \bar{x}_{jk}$$

- Compute an index of *relative frequency* within each group based on presence/absence data.

$$RF_{jk} = \sum_{i=1}^{n_k} b_{ijk} \Big/ n_k$$

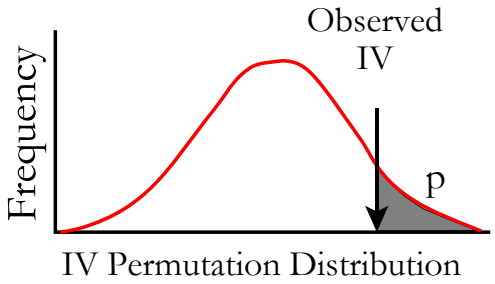
Indicator Species Analysis

- Compute indicator values for each species in each group.
- Each species is assigned as an indicator to the group for which it receives its highest indicator value.
- Indicator values are tested for statistical significance by Monte Carlo permutations of the group membership assignments.

$$IV_{jk} = 100(RA_{jk} \cdot RF_{jk})$$

IV_{jk} = 100 = perfect indication

IV_{jk} = 0 = no indication



Indicator Species Analysis

RELATIVE ABUNDANCE in group, % of perfect indication (average abundance of a given species in a given group over the average abundance of that species groups expressed as a %)

				Group			
			Sequence:	1	2	3	4
			Identifier:	1	2	3	4
			Number of items:	71	36	52	5
	Column	Avg	Max	MaxGrp			
1	AMGO	25	97	4	0	1	2	97
2	AMRO	25	64	4	3	17	16	64
3	BAEA	25	100	3	0	0	100	0
4	BCCH	25	84	4	0	5	10	84
5	BEKI	25	63	3	0	37	63	0
6	BEWR	25	78	4	1	0	21	78
7	BGWA	25	95	3	5	0	95	0
8	BHGR	25	39	4	4	27	29	39
9	BRCR	25	78	2	12	78	10	0
	Averages	24	70		6	20	36	35

RELATIVE FREQUENCY in group, % of perfect indication (% of plots in given group where given species is present)

				Group			
			Sequence:	1	2	3	4
			Identifier:	1	2	3	4
			Number of items:	71	36	52	5
	Column	Avg	Max	MaxGrp			
1	AMGO	28	100	4	0	3	8	100
2	AMRO	62	100	4	18	64	67	100
3	BAEA	0	2	3	0	0	2	0
4	BCCH	34	100	4	1	11	25	100
5	BEKI	4	10	3	0	6	10	0
6	BEWR	21	60	4	1	0	23	60
7	BGWA	5	19	3	1	0	19	0
8	BHGR	75	100	4	28	81	90	100
9	BRCR	24	64	2	18	64	15	0
	Averages	24	49		12	22	26	37

Indicator Species Analysis

INDICATOR VALUES (% of perfect indication, based on combining the above values for relative abundance and relative frequency)

				Group			
			Sequence:	1	2	3	4
			Identifier:	1	2	3	4
			Number of items:	71	36	52	5
	Column	Avg	Max	MaxGrp			
1	AMGO	24	97	4	0	0	0	97
2	AMRO	22	64	4	0	11	11	64
3	BAEA	0	2	3	0	0	2	0
4	BCCH	22	84	4	0	1	3	84
5	BEKI	2	6	3	0	2	6	0
6	BEWR	13	47	4	0	0	5	47
7	BGWA	5	18	3	0	0	18	0
8	BHGR	22	39	4	1	22	26	39
9	BRCR	13	50	2	2	50	2	0
	Averages	11	32		2	9	8	24

MONTE CARLO test of significance of observed maximum indicator value for species 1000 permutations.

	Column	Observed Indicator Value (IV)	IV from randomized groups Mean	S.Dev	p *
1	AMGO	97.3	6.6	4.60	0.0010
2	AMRO	64.5	19.7	6.48	0.0010
3	BAEA	1.9	2.3	2.77	0.5350
4	BCCH	83.8	10.0	5.92	0.0010
5	BEKI	6.1	6.1	4.97	0.3180
6	BEWR	46.7	8.5	5.37	0.0020
7	BGWA	18.2	7.3	5.13	0.0360
8	BHGR	39.2	23.0	5.62	0.0260
9	BRCR	49.6	13.6	5.49	0.0010

* proportion of randomized trials with indicator value equal to or exceeding the observed indicator value.