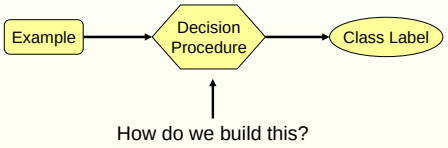
Artificial Intelligence

Knowledge Learning

MONTANA STATE UNIVERSITY

Classification

- ***Def*:** Classification is the task of assigning a label to an example characterizing a set to which that example belongs.



How do we build this?

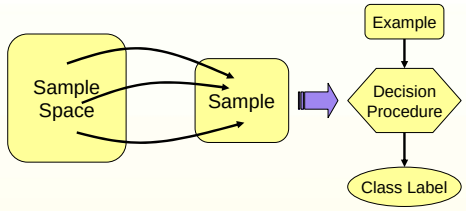
MONTANA STATE UNIVERSITY

Machine Learning

- ***Def*:** Machine learning is adaptation or development of a representation of knowledge from examples for the purpose of solving complex tasks.
- Many classification systems are developed according to the ML model.
- Classification is a central topic in the pattern recognition literature.

MONTANA STATE UNIVERSITY

Learning How to Classify

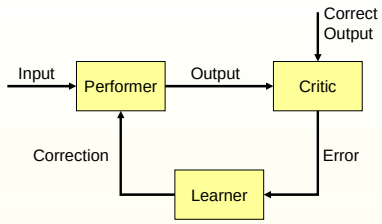


MONTANA STATE UNIVERSITY

4

Mountains Minds

Supervised Learning



MONTANA STATE UNIVERSITY

5

Mountains Minds

The Experiment

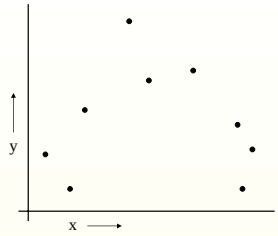
- To the maximum extent possible, the data should be sampled independently (to improve the reliability of comparing the estimates).
  - Select a training set
  - Select a test set
  - Select a revision set (for modifying the hypothesis, if necessary)
- In the event of small populations/samples
  - Training on the entire population/sample, or
  - Select a single set and perform cross-validation

MONTANA STATE UNIVERSITY

6

Mountains Minds

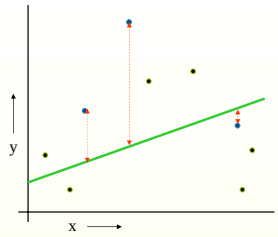
The Test Set Method



- Randomly choose 30% of the data to be in a test set
- The remainder is a training set

MONTANA STATE UNIVERSITY

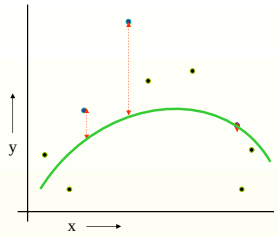
The Test Set Method



(Linear regression example)
Mean Squared Error = 2.4

MONTANA STATE UNIVERSITY

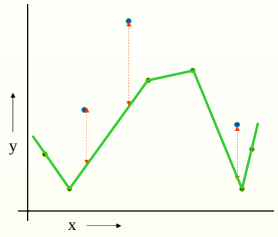
The Test Set Method



(Quadratic regression example)
Mean Squared Error = 0.9

MONTANA STATE UNIVERSITY

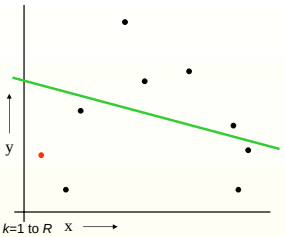
The Test Set Method



(Join the dots example)
Mean Squared Error = 2.2

MONTANA STATE UNIVERSITY

Leave-One-Out Cross Validation

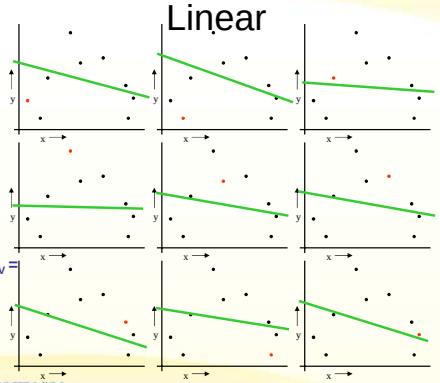


For k=1 to R
1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R–1 datapoints
4. Note your error (x_k, y_k)

When you've done all points, report the mean error.

MONTANA STATE UNIVERSITY

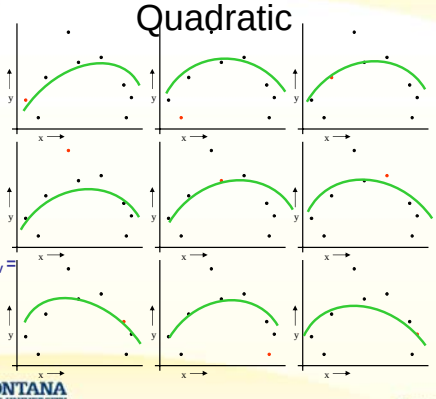
Linear



MSE_{LOOCV} = 2.12

MONTANA STATE UNIVERSITY

Quadratic



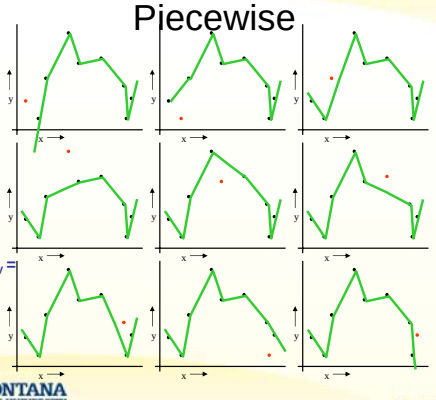
$MSE_{LOOCV} = 0.962$

MONTANA STATE UNIVERSITY

Mountains Minds

13

Piecewise



$MSE_{LOOCV} = 3.33$

MONTANA STATE UNIVERSITY

Mountains Minds

14

K-Fold Cross Validation

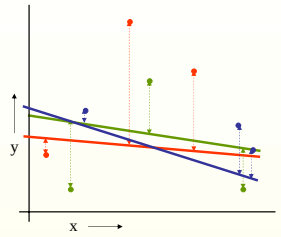
- Randomly break the dataset into *k* partitions (in our example we'll have *k* = 3 partitions colored Red, Green, and Blue)
  - For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.
  - For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.
  - For the blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.
- Then report the mean error.

MONTANA STATE UNIVERSITY

Mountains Minds

15

K-Fold Cross Validation

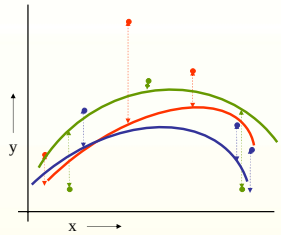


Linear Regression $MSE_{3FOLD} = 2.05$

MONTANA STATE UNIVERSITY

Mountains Minds

K-Fold Cross Validation

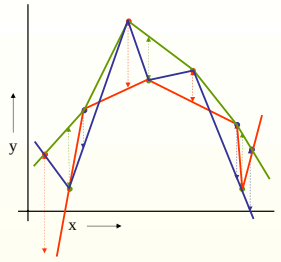


Quadratic Regression $MSE_{3FOLD} = 1.11$

MONTANA STATE UNIVERSITY

Mountains Minds

K-Fold Cross Validation



Piecewise Regression $MSE_{3FOLD} = 2.93$

MONTANA STATE UNIVERSITY

Mountains Minds

Hypothesis Testing

- The following is a "naïve" view of hypothesis testing. There is much more to it than is presented here.
- The most common form of hypothesis test is the "difference of means" test.
- This test determines whether or not once can say that the means of two populations are "different" is a statistically significant way.

MONTANA STATE UNIVERSITY

Mountains Minds

Hypothesis Testing

- Null Hypothesis: The hypothesis that stands unless indicated otherwise statistically.

Sample mean ⟶ $\overline{x} = \mu$ ⟵ Population mean

- Alternative Hypothesis: An alternative to the null hypothesis that might explain the data.

$$\overline{x} \lessgtr \mu$$

MONTANA STATE UNIVERSITY

Mountains Minds

The Key Statistics

Sample mean: $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Sample variance: $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$

Standard error: $SE = \sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}$

t-statistic: $t = \frac{\overline{x}_A - \overline{x}_B}{SE}$

MONTANA STATE UNIVERSITY

Mountains Minds

A Brief t-Statistic Table

df	0.10	0.05	0.025	0.01
5	1.476	2.015	2.570	3.365
10	1.372	1.812	2.228	2.674
15	1.341	1.753	2.131	2.602
20	1.325	1.725	2.086	2.528
25	1.316	1.708	2.059	2.485
30	1.319	1.697	2.042	2.457
∞	1.281	1.645	1.960	2.327

df = degrees of freedom = $n_A + n_B - 2$

MONTANA STATE UNIVERSITY

Mountains Minds

Example

- Suppose we apply 11-fold cross validation on a data set to compare two hypotheses.

$$\bar{x}_A = 0.923 \quad \bar{x}_B = 0.917$$
$$s_A^2 = 0.06 \quad s_A^2 = 0.04$$
$$n_A = 11 \quad n_B = 11$$

Not significant

$$SE = \sqrt{0.0055 + 0.0036} = 0.095$$
$$t = \frac{0.923 - 0.917}{0.095} = 0.063$$

MONTANA STATE UNIVERSITY

Mountains Minds

Concept Learning

- Problem: To infer a description *D* of a target concept *C* in some representation *R* from a set of labeled examples *E*.
- *D* typically represents a Boolean-valued function.
- Example
  - *C* = chair
  - *R* = features describing chairs
  - *D* = conjunction of features
  - *E* = pairs of form {# legs, has-back, …, 1}

MONTANA STATE UNIVERSITY

Mountains Minds

Concept Learning

- Examples E are drawn from some set X.
- Target concept is $c(x):X \rightarrow \{0,1\}$
- Training examples are pairs $\{x,c(x)\}$
- Set of hypotheses, H is used to describes concept c.
- Hypothesis is $h(x):X \rightarrow \{0,1\}$
- Problem of concept learning is to find hypothesis h s.t. $h(x) = c(x)$ over E.
- Note many possible hypotheses may be consistent with E.

MONTANA STATE UNIVERSITY

Mountains Minds

Inductive Learning Hypothesis

- Any hypothesis h that approximates the target concept c of a sufficiently large training set E will also approximate the concept c over unknown examples.
- This hypothesis assumes examples E are drawn from X using some fixed but unknown probability distribution P.
- Unless unseen examples also drawn according to P, no guarantees can be made about any inductive learning algorithm.

MONTANA STATE UNIVERSITY

Mountains Minds

Ordering Hypotheses

- A hypothesis h is said to be more general or equal to another hypothesis h' iff, for all examples x in X, if $h'(x)$ is true (i.e., 1), then $h(x)$ is also true (i.e., 1).
- Note this says nothing about the relationship between h and h' if h' is false (i.e., 0).
- Use $h(x) \geq h'(x)$ to denote that h is more general than h'.
- Defines a partial order of hypotheses.

MONTANA STATE UNIVERSITY

Mountains Minds

Example: EnjoySport

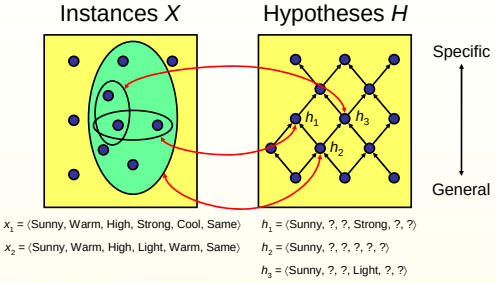
Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

What is a general concept to describe this?

Representing Hypotheses

- There are many possible representations.
- For now, let *h* be a conjunction of constraints on attributes.
- Each constraint can be
  - A specific value (e.g., Water = Warm)
  - A don't care (e.g., Water = ?)
  - No value allowed (e.g., Water = ∅)
- For example,
  - **⟨Sky, Temp, Humid, Wind, Water, Forecst⟩**
  - ⟨Sunny, ?, ?, Strong, ?, Same⟩

Instances to Hypotheses



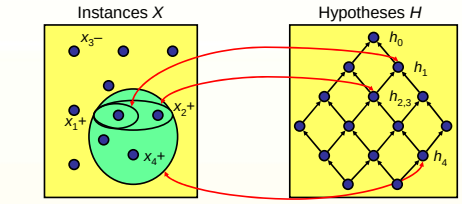
Find-S

- Initialize *H* to the most specific hypothesis in *H*.
- For each *positive* training instance *x*
  - For each attribute constraint a_i in *h* do
    - If the constraint a_i in *h* is satisfied by *x* then do nothing
    - Else replace a_i in *h* by the next more general constraint satisfied by *x*.
- Output hypothesis *h*.

MONTANA STATE UNIVERSITY

Mountains Minds

Hypotheses Space for Find-S



x_1 = ⟨Sunny, Warm, Normal, Strong, Warm, Same⟩, +
x_2 = ⟨Sunny, Warm, High, Strong, Warm, Same⟩, +
x_3 = ⟨Rainy, Cold, High, Strong, Warm, Change⟩, –
x_4 = ⟨Sunny, Warm, High, Strong, Cool, Change⟩, +

h_1 = ⟨S, W, N, S, W, S⟩
h_2 = ⟨S, W, ?, S, W, S⟩
h_3 = ⟨S, W, ?, S, W, S⟩
h_4 = ⟨S, W, ?, S, ?, ?⟩

MONTANA STATE UNIVERSITY

Mountains Minds

Problems with Find-S

- Can't tell whether it has learned the concept.
- Can't tell when the training data is inconsistent.
- Picks only the maximally specific hypothesis.
- Depending on *H*, there might be several hypotheses consistent with *E*.
- Does nothing with negative examples.

MONTANA STATE UNIVERSITY

Mountains Minds

Version Spaces

- Hypothesis h is *consistent* with training examples E iff $h(x) = c(x)$ for each example in E.
  - $Consistent(h,E) \equiv (\forall \langle x,c(x) \rangle \in E)\ h(x) = c(x)$
- The version space $VS(h,E)$ is the subset of all hypotheses in H that are consistent with the training examples E.
  - $VS(h,E) \equiv \{h \in H \mid Consistent(h,E)\}$

MONTANA STATE UNIVERSITY

Mountains Minds

Representing Version Spaces

- Key Idea: Version spaces can be represented by only keeping track of maximally general (G) and maximally specific (S) members of $VS(h,E)$.
- Every member of the version space lies between these boundaries.
  - $VS(h,E) = \{h \in H \mid (\exists s \in S)\ (\exists g \in G)\ (g \geq h \geq s)\}$

MONTANA STATE UNIVERSITY

Mountains Minds

Decision Tree Classification

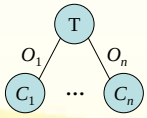
- Let C be a collection of objects
- Let T be a test on an object with outcomes O_1, O_2, …, O_n.
- Each object will have an outcome assigned to it for test T resulting in a partitioning of C into $\{C_1,\ldots,C_n\}$ where each $c_i \in C_i$ has outcome O_i.

MONTANA STATE UNIVERSITY

Mountains Minds

Decision Tree Classification

- This partitioning defines a decision tree structure.
- The partitioning is repeated recursively by applying additional tests on partitions until a consistent partitioning of object space is found.



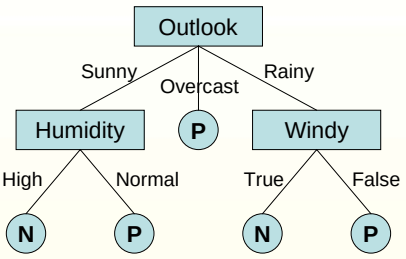
Decision Tree Example

- Suppose we have four features (tests):
  - outlook ∈ {sunny, overcast, rainy}
  - temperature ∈ {cold, mild, hot}
  - humidity ∈ {high, normal}
  - windy ∈ {true, false}
- Keep classifications to two: {Pos, Neg}
- Problem is to "learn" a decision tree from a set of examples and then generalize to unseen examples.

Example Set

No.	Outlook	Temperature	Humidity	Windy	Class
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Sunny	Mild	Normal	True	P
12	Overcast	Mild	High	True	P
13	Overcast	Hot	Normal	False	P
14	Rain	Mild	High	True	N

Example Decision Tree



MONTANA STATE UNIVERSITY

40

Mountains Minds

Efficient Decision Trees

- For efficiency, we want decision trees to be as simple as possible.
- Simple must be defined according to some cost function.
- For now, we will attempt to minimize the number of attribute evaluations

MONTANA STATE UNIVERSITY

41

Mountains Minds

Decision Tree Complexity

- Let **X** = $\{x_1,\ldots,x_n\}$ be a finite set of objects.
- Let **T** = $\{T_1,\ldots,T_t\}$ be a finite set of binary outcome tests.
- Let w be the "cost" of the decision tree (i.e., the expected number of tests required to classify an object x_i).
- Let **DT**(**T**,**X**,w) be the decision problem, "Does there exist a decision tree with cost less than w, given **T** and **X**?"

MONTANA STATE UNIVERSITY

42

Mountains Minds

Decision Tree Complexity

- ***Theorem*: DT(T,X,*w*)** is *NP*-complete.
- ***Proof Idea*:** Show **EC3** $\leq_P$ **DT** where **EC3** is the known *NP*-complete problem of finding an **E**xact **C**over for set **X** where each available subset contains exactly **3** elements.

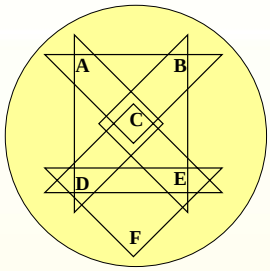
MONTANA STATE UNIVERSITY
Mountains Minds
43

EC3

- ***Instance*:** A finite set **X** with |**X**| = 3*q* (i.e., a multiple of 3) and a collection **C** of 3-element subsets of **X**.
- ***Question*:** Does **C** contain an exact cover of **X**? That is, does there exist a subcollection **C*** $\subseteq$ **C** such that every member of **X** occurs in exactly one member of **C***?

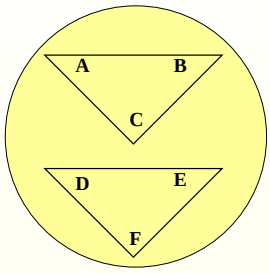
MONTANA STATE UNIVERSITY
Mountains Minds
44

EC3



MONTANA STATE UNIVERSITY
Mountains Minds
45

EC3



MONTANA STATE UNIVERSITY
Mountains Minds
46

Local Search

- Given the complexity of inducing a decision tree, we must use an approximate method based on local search.
- Assumptions:
  - Any correct decision tree for C will classify objects in the same proportion as their representation in C.
  - A decision tree communicates a classification with the expected information necessary to generate a message.

MONTANA STATE UNIVERSITY
Mountains Minds
47

Shannon's Information Theory

- Let T_i = a test on attribute i that partitions C into C_1, …, C_m.

  Let

$$E(T_i) = \sum_{j=1}^{m_i} \frac{p_i^j + n_i^j}{p+n} I(p_i^j, n_i^j)$$

  where

$$I(p,n) = -\frac{p}{p+n}\lg\frac{p}{p+n} - \frac{n}{p+n}\lg\frac{n}{p+n}$$

MONTANA STATE UNIVERSITY
Mountains Minds
48

Information Gain

- $I(p,n)$ is the amount of information contained in the classification structure, sometimes referred to as "entropy."
- $E(T_i)$ is the expected information contained in the subtree with test T_i at the root.
- Observe that $(p_i^j + n_i^j)/(p+n)$ is the probability that object c_k belongs to partition C_j given outcome j for test T_i.
- Information gain is given by
  - $gain(T_i) = I(p,n) - E(T_i)$
- We want to choose T_i such that $gain(T_i)$ is maximized.

MONTANA STATE UNIVERSITY
Mountains Minds
49

General Form

- These equations can be generalized for k classes:

  Let

$$E(T_i) = \sum_{j=1}^{m_i} \frac{c_i^{1,j} + \cdots + c_i^{k,j}}{c_1 + \cdots + c_k} I(c_i^{1,j}, \ldots, c_i^{k,j})$$

  where

$$I(c_1, \ldots, c_k) = -\sum_{i=1}^{k} \frac{c_i}{c_1 + \cdots + c_k} \lg \frac{c_i}{c_1 + \cdots + c_k}$$

MONTANA STATE UNIVERSITY
Mountains Minds
50

Example

- Previous example had 14 objects, 9 of which were positive and 5 negative.
- We start by computing $I(p,n)$.

$$I(p,n) = -\frac{9}{14}\lg\frac{9}{14} - \frac{5}{14}\lg\frac{5}{14} = 0.94$$

MONTANA STATE UNIVERSITY
Mountains Minds
51

Example

- Next we compute the weights for the outcome of each attribute of each test.
- For illustration, consider the test *outlook*.

$$I_{\text{sunny}}(p,n) = -\frac{2}{5}\lg\frac{2}{5} - \frac{3}{5}\lg\frac{3}{5} = 0.971$$

$$I_{\text{overcast}}(p,n) = -\frac{4}{4}\lg\frac{4}{4} - \frac{0}{4}\lg\frac{0}{4} = 0.0$$

$$I_{\text{rainy}}(p,n) = -\frac{3}{5}\lg\frac{3}{5} - \frac{2}{5}\lg\frac{2}{5} = 0.971$$

MONTANA STATE UNIVERSITY

Mountains Minds

Example

- From this, we compute *E*(Outlook)

$$E(\text{Outlook}) = \frac{5}{14} I_{\text{sunny}}(p,n) + \frac{4}{14} I_{\text{overcast}}(p,n) + \frac{5}{14} I_{\text{rainy}}(p,n) = 0.694$$

MONTANA STATE UNIVERSITY

Mountains Minds

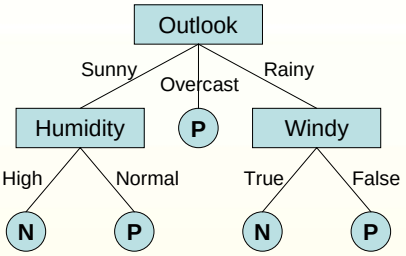
Example

- Similarly
  - *E*(Temperature) = 0.911
  - *E*(Humidity) = 0.789
  - *E*(Windy) = 0.892
- Now for information gain
  - *gain*(Outlook) = 0.246 ← Choose this one.
  - *gain*(Temperature) = 0.029
  - *gain*(Humidity) = 0.151
  - *gain*(Windy) = 0.048

MONTANA STATE UNIVERSITY

Mountains Minds

Final Tree



MONTANA STATE UNIVERSITY

55

Mountains Minds

Appropriate Problems for DT

- Instances represented by attribute-value pairs.
- Target function has discrete output values (otherwise, requires a *regression* tree).
- Disjunctive descriptions might be required (traditional candidate elimination fails).
- Training data may contain errors.

MONTANA STATE UNIVERSITY

56

Mountains Minds

Issues with DT Learning

- When should we stop growing the tree?
  - Need to avoid overfitting training data.
- What about attributes with continuous values (e.g., temperature)?
- What about discrete attributes with many values (e.g., birthday)?
- What if attribute values are sometimes unknown (e.g., diagnostic test result)?
- What if attributes have different costs (e.g., simple observation vs. lab test)?

MONTANA STATE UNIVERSITY

57

Mountains Minds

Overfitting

- As with other methods, the goal of DT is to generalize based on limited examples to classify previously unseen examples with minimal error.
- A decision tree, *d*, is said to *overfit* the training data if there exists some other tree, *d'*, such that *d* has smaller error than *d'* over the training examples, but *d'* has smaller error than *d* over the entire distribution of instances.

MONTANA STATE UNIVERSITY
Mountains & Minds
58

Halting or Pruning

- Overfitting can be reduced by
  - changing termination criterion to halt earlier, or
  - allowing the tree to overfit and then post-prune.
- Criterion to use
  - Optimize performance on separate pruning set.
  - Include only statistically significant tree splits.
  - Minimize (*size*[*T*] + *size*[misclassified points]).

MONTANA STATE UNIVERSITY
Mountains & Minds
59

Stopping Policies

- A stopping policy indicates when further growth of a tree at node *t* is counterproductive.
  - All examples are of the same class (pure).
  - The attribute values of all examples are the same.
  - All examples have missing values.
  - At most one class has a number of examples larger than a user-specified threshold.
  - Subtree increases classification error on validation set.

MONTANA STATE UNIVERSITY
Mountains & Minds
60

Reduced-Error Pruning

- Split data into training and pruning (sometimes called validation) sets.
- Do until further pruning is harmful.
  - Evaluate impact on pruning set of pruning each possible node (plus those below).
  - Greedily remove the one that improves pruning set accuracy.

MONTANA STATE UNIVERSITY

Mountains Minds

Continuous Attributes

- Consider an attribute $A_i \in$ [*low*, *high*]
- In training set, there will only be a finite number of values.
- Sort the examples by these values
  - $V_1, V_2, \ldots, V_n$
- Consider potential splits values as midpoints between adjacent values (i.e., $[V_j + V_{j+1}] / 2$)
- Split on test $A_i < [V_j + V_{j+1}] / 2$

MONTANA STATE UNIVERSITY

Mountains Minds

Attributes with Many Values

- Binary splits tend to make trees more difficult to interpret.
- We could discretize real valued attributes using split points.
- This results in attributes with several discrete values.
- Some attributes naturally have large number of discrete values.
- Information gain tends to favor attributes with a large number of values.

MONTANA STATE UNIVERSITY

Mountains Minds

Gain-Ratio Criterion

- Answers the question, "What is the worth of attribute A independent of values?"
- Inquiring about an attribute's worth, independent of a specific test, also acquires information.
- Maximize the gain ratio.

$$IV(A) = -\sum_{i=1}^{m} \frac{p_i + n_i}{p+n} \lg \frac{p_i + n_i}{p+n}$$

$$gain\text{-}ratio(A) = gain(A) / IV(A)$$

MONTANA STATE UNIVERSITY

64

Mountains Minds

Other "Impurity" Measures

- When we select an attribute to partition the data space, we select a hyperplane to divide the space.
- Let *H* be a hyperplane corresponding to a split along a particular attribute.
- Assume there are only two classes of examples.
- ***Def:*** If all examples in a partition fall in the same class, then that partition is *homogeneous*.

MONTANA STATE UNIVERSITY

65

Mountains Minds

Other "Impurity" Measures

- Within a particular partition, also consider the subsets of that partition, corresponding to members of the various classes.
- Let L_i = # instances in category *i* on left
- Let R_i = # instances in category *i* on right.
- In a partition, the subsets of the smaller size are the *minority*.

$$MinorityL = \sum_{i=1, i \neq \max L_i}^{k} L_i$$

- In a partition, the subsets of the greater size are the *majority*.

$$MinorityR = \sum_{i=1, i \neq \max R_i}^{k} R_i$$

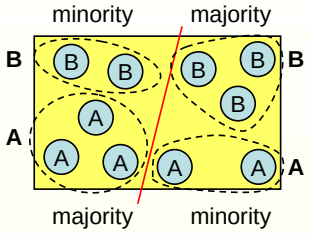
- Minimize impurity.

MONTANA STATE UNIVERSITY

66

Mountains Minds

Example



MONTANA STATE UNIVERSITY
Mountains Minds
67

Other "Impurity" Measures

- Max Minority (MM)
  - *MM*(*A*) = max{*MinorityL*, *MinorityR*}
- Sum Minority (SM)
  - *SM*(*A*) = *MinorityL* + *MinorityR*
- Sum of Impurity/Variances (two-class)
  - Let $P_1,\ldots,P_L$ be examples on left partition.
  - Let $C_P \in \{0,1\}$ be class label for point P.
  - T_L = Subset of examples to left (similar for T_R).

$$avg = \frac{1}{|T_L|}\sum_{i=1}^{|T_L|} C_{P_i},\ imp(T_L) = \sum_{i=1}^{|T_L|}\left(C_{P_i} - avg\right)^2$$
$$SI(T) = imp(T_L) + imp(T_R)$$

MONTANA STATE UNIVERSITY
Mountains Minds
68

Oblique Decision Trees

- Decision trees where each node splits on a single attribute are called "axis-parallel" decision trees.
- Many concepts do not have decision boundaries parallel with attribute axes.
- A variation on the axis-parallel decision tree, called the oblique decision tree, permits linear splits not parallel with attribute axes.

MONTANA STATE UNIVERSITY
Mountains Minds
69

Oblique Splits

- Let an example take the form $X = x_1, x_2, \ldots, x_d; C_j$, where C_j is the class label and the x_i's are real-valued attributes.
- The test at each node of the decision tree will then have the form

$$\sum_{i=1}^{d} a_i x_i + a_{d+1} > 0$$

- $a_1, \ldots, a_{d+1}$ are real-valued coefficients.

MONTANA STATE UNIVERSITY

Mountains Minds

Oblique Classifier 1 (OC1)

- Select a split using information gain for a "best" axis-parallel split.
- Select a linear combination of attributes at random.
- Perturb coefficients as long as impurity is reduced.
- Select the split with minimum impurity.

MONTANA STATE UNIVERSITY

Mountains Minds

Perturbation

- Consider a coefficient a_m.
- Treat that coefficient as a variable and all others as constants.
- $V_j = a_{d+1} + \Sigma\, a_i x_i > 0$ is a function of a_m.
- Then $a_m > [a_m x_{jm} - V_j] / x_{jm} =_{\text{df}} U_j$ indicates example above hyperplane.
- Compute V_j using midpoint technique. Then compute all U_j using above.
- Sort all U_j in non-decreasing value.
- Select new a_m' as best univariate split.
- Accept improvement. Accept decline with prob. (Similar to simulated annealing)

MONTANA STATE UNIVERSITY

Mountains Minds