Induction of Decision Trees (part 2)

Andrea Danyluk
April 12, 2017

Announcements

- Classifier learning assignment posted
  - Base assignment + 1-2 extensions
- Final project
  - Discuss ideas with me this week.
  - Schedule/deliverables posted on the course website.

Final Projects

- Past projects
  - Scrabble
  - Automatic file organization into folders https://www.meta.sc/
  - Bidding Tic-Tac-Toe
  - Voice Recognition
  - Cross-lingual lexical substitution
  - Facebook status generation
  - 9x9 Go
  - Othello player (trained through a combination of GAs and NNs)
  - Evolution of "optimal" Darwin creatures
  - Reproduce experiments in a paper comparing novel search algorithms

Today's Lecture

- Classifier learning: finishing up with decision trees

Decision Trees on Real Problems

- How do we assess a decision tree's performance?
- How do we handle attributes with numeric values?
- Missing attribute values?
- How do we handle noise?
- Bias in attribute selection?

Assessing Performance

- Performance task is to predict the classes of unseen examples.
- Assessing the quality of the decision tree involves checking its classifications of labeled test examples.
- Requires that we leave some of our data out of the training set, so that we can test with it.

Test Methodology

- Collect a large set of examples.
  - If large enough (rare, even with today's large data sets), can simply divide it into a training set and a test set.
- More often: Divide it into k disjoint sets.
  - Say k = 10.

  Reserve set 1 for testing. Train on rest.

  Reserve set 2 for testing. Train on rest.

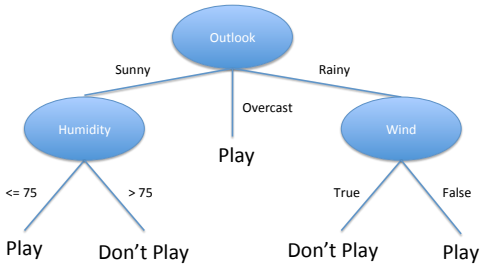
  …

  Reserve set 10 for testing. Train on rest.
- Compute average accuracy over the k "folds".
- k-fold cross-validation.
- Really want stratified cross-validation. Shuffling the data before dividing into folds is a low-cost method to approximate (but not necessarily guarantee) stratification.

Attributes with Numeric Values: the Golf Tree

	Outlook	Temp	Humidity	Wind	Class
Example1	Sunny	85	85	False	Don't Play
Example2	Sunny	80	90	True	Don't Play
Example3	Overcast	83	88	False	Play
Example4	Rainy	70	96	False	Play
Example5	Rainy	68	80	False	Play
Example6	Rainy	65	70	True	Don't Play
Example7	Overcast	64	65	True	Play
Example8	Sunny	72	95	False	Don't Play
Example9	Sunny	69	70	False	Play
Example10	Rainy	75	80	False	Play
Example11	Sunny	75	70	True	Play
Example12	Overcast	72	90	True	Play
Example13	Overcast	81	75	False	Play
Example14	Rainy	71	96	True	Don't Play

"Is it a good day to play golf?"



Attributes with Numeric Values

- Split the numeric range into two groups:

  values <= threshold

  values > threshold
- How to select the threshold:
  - Sort the examples by the values of the attribute.
  - Search the examples, noting adjacent examples that belong to different classes.
  - The average values at the transition points represent potential splits.
  - Evaluate each split by applying the information gain formula.
  - Choose the best.
- Compare the gain for the best split against information gain for the remaining attributes.

Attributes with Numeric Values: the Golf Tree

Considering only the examples with Outlook=Sunny

	Humidity	Class
Example9	70	Play
Example11	70	Play
Example1	85	Don't Play
Example2	90	Don't Play
Example8	95	Don't Play

Only one split point here, but in general working with real-valued variables can be quite expensive.

Attributes with Numeric Values: the Golf Tree

Considering only the examples with Outlook=Sunny

	Humidity	Class
Example9	70	Play
Example11	70	Play
Example1	85	Don't Play
Example2	90	Don't Play
Example8	95	Don't Play

But wait: Shouldn't 77.5 be the split point? The tree says 75. Some learners take the value over the entire training set closest to the split point.

Attributes with Numeric Values vs Nominal

- Nominal attributes: Never consider the attribute a second time along a path in the tree.
- Real-valued attributes?

Attributes with Missing Values

- Say that a given training example has a missing attribute value.
- How do we handle it when trying to compute the information gain of the attribute at a node *n* in the tree?
- Assign it a value for the purposes of calculation.
  - Use the value that is most common among training examples at node *n*. (or simply the value that is most common among all examples in the training set)
  - Use the value that is most common among examples at *n* that have the same classification. (or do this over the entire training set)
  - Treat the example as if it can be divided into fractions according to the frequency of the possible attribute values over all examples at node *n*.

Attributes with Missing Values

- Say that a given training example has a missing attribute value.
- C4.5:
  - When computing information gain, ignore examples with missing attributes.
  - When splitting the examples at a node during tree-building:
    - If an example is missing a value for that attribute, send a fraction of it to each child.
  - When testing, send a fraction to each child at a split for an attribute with a missing value. Collect all possible classifications and decide among them.