Classifier Learning:
Induction of Decision Trees

Andrea Danyluk
April 10, 2017

Announcements

- Programming Assignment 4: Filtering
  - Due tomorrow.
  - Will send out the link for code review sign-up. If you haven't done two code reviews, please sign up.
- Final project
  - Discuss ideas with me this week.
  - Will post the full schedule/deliverables on Wednesday.

Today's Lecture

- Classifier learning: decision trees
- Note that the original syllabus said neural nets first. Switching the order.

Machine Learning includes…

- Learning how to do something
- Learning how to do something better
- Learning new facts
- …

Supervised Classifier Learning

- In the category of "learning new facts"
- **Inductive**
  - Algorithm induces a general rule (or set of general rules) from a set of observed instances
  - No explicit background knowledge about the domain of application
- **Supervised**
  - Given a set of training examples (**x**, y), where **x** is a feature vector describing an example and y is its class

Inductive = Knowledge-free?

- A possible claim: inductive classifier learners make no use of explicit background knowledge about the domain
- Not exactly: the attributes describing the examples are provided
  - Feature engineering is non-trivial

Inductive Bias

- The learned representation is set by the algorithm
- How the training examples are used is determined by the algorithm
- Many other ways in which the learning is influenced

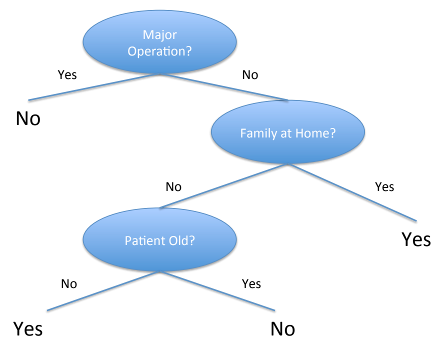
- Any preference for one hypothesis over another, beyond mere consistency with the examples, is called a **bias**

Send patient home from hospital post-op?

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

Learn a classifier that, given a new patient, will determine whether the patient should be sent home or not.

Decision tree for "Send patient home post-op?"



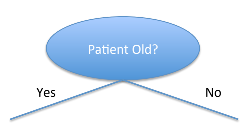
TDIDT: Top down induction of decision trees

If all examples are from the same class
  The tree is a leaf with that class name
Else
  Pick an attribute for the decision node
  Construct one edge for each possible value of that attribute
  Partition examples by attribute value
  Build subtrees recursively

Note that this is a ***greedy*** *algorithm*

Selecting attributes on which to split

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	No	Yes



Which is better?

Characteristics of Attribute Tests

- Let Y be the set of examples of class "Yes, send home"
  Let N be the set of examples of class "No"
- Say that $|Y| = 10$, $|N| = 10$
- Say that all of our attributes are Boolean.
- A test at any non-leaf node splits the data into two subsets, T_1 and T_2
  – The **best test** is one that produces $T_1 = Y$, $T_2 = N$.
  – The **worst test** is one such that T_1 contains an equal share of Y and N and T_2 does as well.

Entropy

A measure of the disorder/impurity of a set of examples.

- Let T be our set of training examples.
- Let $C_1, C_2, \ldots, C_n$ be the class labels assigned to examples in T.
- Let freq(C_i, T) be the number of examples in the training set that belong to class C_i.
- Let |T| be the number of examples in the training set.

Entropy(T) = - Σ_i(freq(C_i,T)/|T|) * $\log_2$ (freq(C_i,T)/|T|)

Just one way to think about the entropy measure

- Say I have a bag of 100 marbles.
  - 99 are blue
  - 1 is red
- If I pull out a marble and announce that it's blue, that's not very informative.

  -$\log_2$ (freq(C_i,T)/|T|) bits

  High probability corresponds to low information
- If I pull out a marble and announce that it's red, that's much more interesting, but it will only happen 1/100 of the time.

Information Gain

- Select the test that decreases entropy most.
- Let X be an attribute.
  - Say that X is discrete-valued and has n possible values.
  - If X were selected as a test, we would create a decision node with n branches.
- Let j be a possible value of X. Let T_j be the examples that have value j for attribute X.
- We can compute the average entropy that results from making this split:

$$\text{Entropy}_X(T) = \Sigma_j(|T_j|/|T| * \text{Entropy}(T_j))$$

$$\text{Gain}(T, X) = \text{Entropy}(T) - \text{Entropy}_X(T)$$

Choose the attribute with the greatest gain.

Building the hospital-release tree

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

Entropy = - (3/5 $\log_2$ 3/5 + 2/5 $\log_2$ 2/5)

= 0.6 * .74 + 0.4 * 1.32

= .972

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

Entropy = .972

$\text{Entropy}_{\text{MajorOperation}}$

MajorOperation=Yes: Entropy = 0

MajorOperation=No: - (1/3 $\log_2$ 1/3 + 2/3 $\log_2$ 2/3)

= .9042

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

Entropy = .972

$\text{Entropy}_{\text{MajorOperation}}$

= 2/5 * 0 + 3/5 * .9042

= .54

Gain = .432

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

Entropy = .972

$\text{Entropy}_{\text{FamilyAtHome}}$

FamilyAtHome=Yes: Entropy = 0
FamilyAtHome=No: $- (1/4 \log_2 1/4 + 3/4 \log_2 3/4)$
= .81

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

Entropy = .972

$\text{Entropy}_{\text{FamilyAtHome}}$

= 1/5 * 0 + 4/5 * .81
= .648

Gain = .324

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

Entropy = .972

$\text{Entropy}_{\text{Old}}$

Old=Yes: Entropy = $-(1/3 \log_2 1/3 + 2/3 \log_2 2/3)$ =.9042
Old=No: $- (1/2 \log_2 1/2 + 1/2 \log_2 1/2)$ = 1

Major Operation?	Family at Home?	Old?	Send Home?
Yes	No	Yes	No
Yes	No	No	No
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

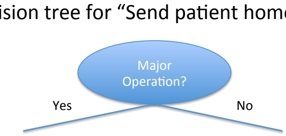
Entropy = .972

$\text{Entropy}_{\text{Old}}$

= 3/5 *.9042 + 2/5 * 1
= .942

Gain = .03

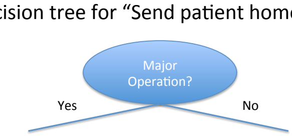
Decision tree for "Send patient home post-op?"



No

Major Operation?	Family at Home?	Old?	Send Home?
No	No	Yes	No
No	Yes	Yes	Yes
No	No	No	Yes

Decision tree for "Send patient home post-op?"

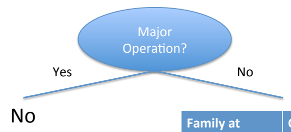


No

Family at Home?	Old?	Send Home?
No	Yes	No
Yes	Yes	Yes
No	No	Yes

Entropy = $-(1/3 \log_2 1/3 + 2/3 \log_2 2/3)$
=.9042

Decision tree for "Send patient home post-op?"



Family at Home?	Old?	Send Home?
No	Yes	No
Yes	Yes	Yes
No	No	Yes

$\text{Entropy}_{\text{FamilyAtHome}}$
= 1/3 * 0 + 2/3 * 1
= .67

Entropy =.9042

Gain = .234

Decision tree for "Send patient home post-op?"

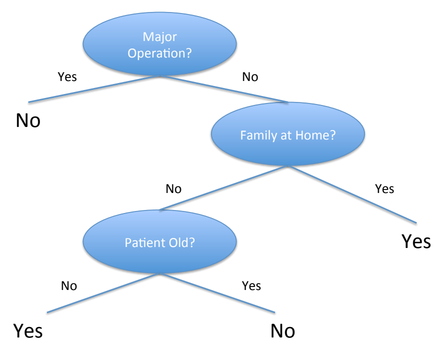
Family at Home?	Old?	Send Home?
No	Yes	No
Yes	Yes	Yes
No	No	Yes

$\text{Entropy}_{\text{Old?}}$
= 1/3 * 0 + 2/3 * 1
= .67

Entropy =.9042

Gain = .234

Decision tree for "Send patient home post-op?"



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.