DM I: Block 'Classification'

Unit 'Decision Trees'

Myra Spiliopoulou

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FACULTY OF COMPUTER SCIENCE

Materials

- Algorithms and equations: Chapter 3 of the course book
- Pictures: Chapter 4 of the 1st edition, but with pointers to the course book

Tree Induction Algorithms

DT for verterbrate classification

Tan et al., Ch.4 (2006) [a]

[a] In the book of the course, this is Figure 3.4

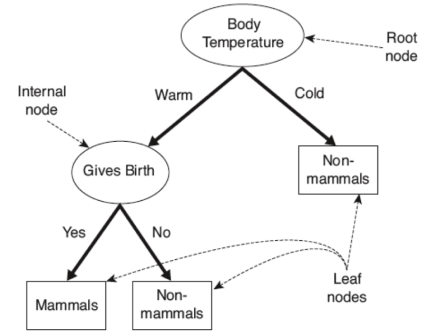

Figure 4.4. A decision tree for the mammal classification problem.

Tree Induction Algorithms

Using the DT for verterbrate classification Tan et al., Ch.4 (2006) [a]

[a] In the book of the course, this is Figure 3.5

to classify a flamingo:

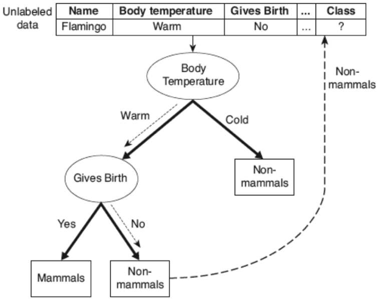

Figure 4.5. Classifying an unlabeled vertebrate. The dashed lines represent the outcomes of applying various attribute test conditions on the unlabeled vertebrate. The vertebrate is eventually assigned to the `Non-mammal` class.

Tree Induction Algorithms

Learning decision trees:

- A very old, simple tree induction algorithm: Hunt's algorithm
- Split criteria for decision tree learners
- Learning bushy classifiers
- Learning binary classifiers

on the example of the patient responses' dataset:

Id	I2	I15	I30	I22	I31	I9	I26	Response
#1	f	VH	yes	better	no	r	no	yes
#2	m	M	no	better	no	b	yes	no
#3	m	M	no	worse	no	b	no	no
#4	f	VH	yes	worse	no	b	no	yes
#5	m	L	no	no effect	no	l	no	no
#6	m	M	no	better	no	l	no	no
#7	f	VH	yes	better	yes	l	yes	yes
#8	f	H	no	better	yes	r	no	no
#9	f	H	yes	better	no	l	no	yes
#10	m	M	yes	worse	no	b	no	no
#11	m	M	no	no effect	no	l	no	no
#12	f	H	no	no effect	no	r	no	no
#13	f	L	yes	better	no	l	yes	yes
#14	m	M	no	worse	no	b	no	no
#15	m	L	no	no effect	no	l	yes	no

Tree Induction Algorithms - Hunt's algorithm

Hunt's algorithm based on [Ch 3, Section 3.3.1]

INPUT: Training set D, Labelset $L = \{y_1, \ldots, y_k\}$

At the current node v, invoke
$hunt(v, L)$

IF exists $y \in L$ such that $\forall x \in v : label(x) = y$

THEN do

1. set y as the label of the whole v
2. return

ELSE do

1. compute $children(v)$ by invoking $split(v, L)$
2. for each $u \in children(v)$
   $hunt(u, L)$

Tree Induction Algorithms - Split functions

EXAMPLE: Candidate splits on the 'patient responses' dataset

- Split on I2:
  - **I2=f** [yes:(#1,#4,#7,#9,#13), no:(#8,#12)]
  - **I2=m** [yes: (), no: (#2,#3,#5,#6,#10, #11, #14, #15)]
- Split on I22:
  - **I22=better** [yes: (#1,#7, #9, #13), no: (#2,#6,#8)]
  - **I22=worse** [yes: (#4), no: (#3,#10, #14)]
  - **I22=no effect** [yes: (), no: (#5, #11, #12, #15)]

Tree Induction Algorithms - Split functions

EXAMPLE: Candidate splits on the 'patient responses' dataset

- Split on I2:
  - **I2=f** [yes:(#1,#4,#7,#9,#13), no:(#8,#12)]
  - **I2=m** [yes: (), no: (#2,#3,#5,#6,#10, #11, #14, #15)]
- Split on I22:
  - **I22=better** [yes: (#1,#7, #9, #13), no: (#2,#6,#8)]
  - **I22=worse** [yes: (#4), no: (#3,#10, #14)]
  - **I22=no effect** [yes: (), no: (#5, #11, #12, #15)]

Which split is better ?

Tree Induction Algorithms - Split functions

EXAMPLE: Candidate splits on the 'patient responses' dataset

- Split on I2:
  - **I2=f** [yes:(#1,#4,#7,#9,#13), no:(#8,#12)]
  - **I2=m** [yes: (), no: (#2,#3,#5,#6,#10, #11, #14, #15)]
- Split on I22:
  - **I22=better** [yes: (#1,#7, #9, #13), no: (#2,#6,#8)]
  - **I22=worse** [yes: (#4), no: (#3,#10, #14)]
  - **I22=no effect** [yes: (), no: (#5, #11, #12, #15)]

Which split is better ?

Typical objectives for the split function

- Minimize the impurity with respect to the target variable
- Minimize the misclassification rate

Tree Induction Algorithms - Split functions

Let D be the training set, $v \subseteq D$ be a tree node, and $L = \{y_1, \ldots, y_k\}$ be the set of labels. [1]

Example split functions

$$MisclassificationRate(v) = 1 - \max_{y \in L} p(y|v)$$

[1] cf. Equations 3.4, 3.5, 3.6 – notice the differences in notation

Tree Induction Algorithms - Split functions

Let D be the training set, $v \subseteq D$ be a tree node, and $L = \{y_1, \ldots, y_k\}$ be the set of labels. [1]

Example split functions

$$MisclassificationRate(v) = 1 - \max_{y \in L} p(y|v)$$

$$Gini(v) = 1 - \sum_{y \in L} p(y|v)^2$$

[1] cf. Equations 3.4, 3.5, 3.6 – notice the differences in notation

Tree Induction Algorithms - Split functions

Let D be the training set, $v \subseteq D$ be a tree node, and $L = \{y_1, \ldots, y_k\}$ be the set of labels. [1]

Example split functions

$$MisclassificationRate(v) = 1 - \max_{y \in L} p(y|v)$$

$$Gini(v) = 1 - \sum_{y \in L} p(y|v)^2$$

$$entropy(v) = -\sum_{y \in L} p(y|v) \log p(y|v)$$

where $0 \log 0$ is defined to be zero.

[1] cf. Equations 3.4, 3.5, 3.6 – notice the differences in notation

Tree Induction Algorithms - Split functions

Let D be the training set, $v \subseteq D$ be a tree node, and $L = \{y_1, \ldots, y_k\}$ be the set of labels.

$$\begin{aligned} MisclassificationRate(v) &= 1 - \max_{y \in L} p(y|v) \\ Gini(v) &= 1 - \sum_{y \in L} p(y|v)^2 \\ entropy(t) &= -\sum_{y \in L} p(y|v) \log p(y|v) \end{aligned}$$

where $0 \log 0$ is defined to be zero.

Tree Induction Algorithms - Split functions

Let D be the training set, $v \subseteq D$ be a tree node, and $L = \{y_1, \ldots, y_k\}$ be the set of labels.

$$\begin{aligned} MisclassificationRate(v) &= 1 - \max_{y \in L} p(y|v) \\ Gini(v) &= 1 - \sum_{y \in L} p(y|v)^2 \\ entropy(t) &= -\sum_{y \in L} p(y|v) \log p(y|v) \end{aligned}$$

where $0 \log 0$ is defined to be zero.

How do these functions differ in their behaviour?

- Compute the values of the split functions for these nodes [a]:
  - v_1 : 0 members with label Y, 6 members with label N
  - v_2 : 1 member with label Y, 5 members with label N
  - v_3 : 3 members with label Y, 3 members with label N
- Compute the values of the split functions for the 7 attributes in the 'patients responses' dataset

[a] Examples from Tan et al., Ch.4 (2006)

Tree Induction Algorithms - Split functions

Behaviour of the split functions for binary classification [a]

[a] In the book of the course, this is Figure 3.11

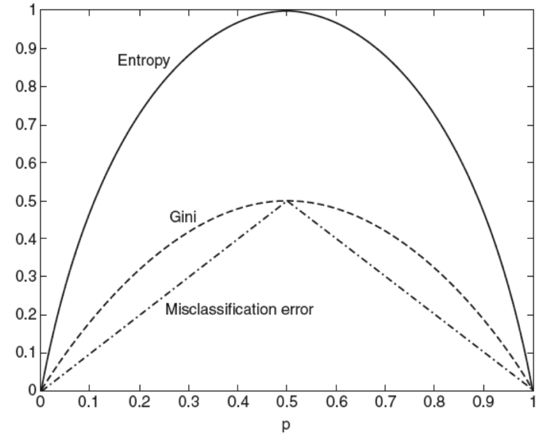

Figure 4.13. Comparison among the impurity measures for binary classification problems.

More on splits 1/2

RECALL example:

EXAMPLE: Candidate splits on the 'patient responses' dataset

- Split on I2:
  - **I2=f** [yes:(#1,#4,#7,#9,#13), no:(#8,#12)]
  - **I2=m** [yes: (), no: (#2,#3,#5,#6,#10, #11, #14, #15)]
- Split on I22:
  - **I22=better** [yes: (#1,#7, #9, #13), no: (#2,#6,#8)]
  - **I22=worse** [yes: (#4), no: (#3,#10, #14)]
  - **I22=no effect** [yes: (), no: (#5, #11, #12, #15)]

More on splits 1/2

RECALL example:

EXAMPLE: Candidate splits on the 'patient responses' dataset

- Split on I2:
  - **I2=f** [yes:(#1,#4,#7,#9,#13), no:(#8,#12)]
  - **I2=m** [yes: (), no: (#2,#3,#5,#6,#10, #11, #14, #15)]
- Split on I22:
  - **I22=better** [yes: (#1,#7, #9, #13), no: (#2,#6,#8)]
  - **I22=worse** [yes: (#4), no: (#3,#10, #14)]
  - **I22=no effect** [yes: (), no: (#5, #11, #12, #15)]

Two ways of splitting an attribute that takes $<< 2$ values

- ⋆ **multi-split:** one child node per attribute value
- ⋆ **binary split:** pick one value zz and split into two child nodes, one for zz and one for 'not zz'

Two types of classifiers – bushy & binary

More on splits 2/2

How to split a node v on a *continuous* attribute a?

Tree Induction Algorithms - Dealing with non-categorical attributes

Order preserving n-split of a node v on an attribute a that takes continuous values:

- *Greedy way:*
  Iteratively consider candidate positions within the valuerange of a, e.g.
  - by sampling n values randomly for some n (input parameter)
- *Discretization:*
  - in an unsupervised way, e.g. by
    building n homogeneous and well-separated clusters, or
    by building a histogramm of equisized bins
  - in a supervised way, e.g. by partitioning the data so as to minimize the impurity measure

  acquiring n groups of instances [2].

This results in a candidate split of of node v to n children, i.e. to a set $children(v, a)$, for which we can then compute the gain on inpurity.

[2] The number of groups n may be an input parameter or may be derived.

Tree Induction Algorithm - Quinlan's ID3 (simplified)

Learning a bushy classifier with ID3

INPUT: Training set D, Labelset $L = \{y_1, \ldots, y_k\}$, set of attributes A

At the current node v, invoke
$ID3(v, L, A)$

IF $\exists y \in L$ such that for most of the $x \in v : label(x) = y$ THEN do

set y as the label of the whole v and return

ELSE IF $A = \emptyset$ THEN do

1. identify the majority class label of v, y
2. set y as the label of the whole v and return

ELSE do

1. set a_{best} as the $\arg\max_{a \in A}\{InfGain(v, L, a)\}$
2. compute $children(v, a_{best})$ by splitting v on the values of a_{best}
3. for each $u \in children(v, a_{best})$, invoke $ID3(u, L, A \setminus \{a_{best}\})$

Tree Induction Algorithms - Entropy and Information Gain

Gain on impurity

For a node v split on the values of attribute $a \in A$ into $children(v,a)$, the gain of this split is:

$$\Delta(v,a) = I(v) - \sum_{u \in children(v,a)} \frac{|u|}{|v|} I(u)$$

where $I(\cdot)$ is an impurity measure [3].

[3] See description of gain around Equations 3.7, 3.8 (different notation)

[4] NOTE: L is fixed, so $InfGain(v,a) \equiv InfGain(v,L,a)$.

Tree Induction Algorithms - Entropy and Information Gain

Gain on impurity

For a node v split on the values of attribute $a \in A$ into $children(v,a)$, the gain of this split is:

$$\Delta(v,a) = I(v) - \sum_{u \in children(v,a)} \frac{|u|}{|v|} I(u)$$

where $I(\cdot)$ is an impurity measure [3]. If we set

$$I(v) := entropy(v) = -\sum_{y \in L} p(y|v) \log p(y|v)$$

then $\Delta(v,a) \equiv InfGain(v,a)$. [4]

Which attribute gives the best root split in the 'patient responses' dataset?

[3] See description of gain around Equations 3.7, 3.8 (different notation)

[4] NOTE: L is fixed, so $InfGain(v,a) \equiv InfGain(v,L,a)$.

Tree Induction Algorithms - Information Gain and Gain Ratio

For a node v split on the values of attribute $a \in A$ into $children(v,a)$ ...

Information Gain

$$InfGain(v,a) = entropy(v) - \sum_{u \in children(v,a)} \frac{|u|}{|v|} entropy(u)$$

Tree Induction Algorithms - Information Gain and Gain Ratio

For a node v split on the values of attribute $a \in A$ into $children(v,a)$ ...

Information Gain

$$InfGain(v,a) = entropy(v) - \sum_{u \in children(v,a)} \frac{|u|}{|v|} entropy(u)$$

Intrinsic Information from Piatesky-Shapiro

$$IntrinsicInformation(v,a) = - \sum_{u \in children(v,a)} \frac{|u|}{|v|} \log \frac{|u|}{|v|}$$

Tree Induction Algorithms - Information Gain and Gain Ratio

For a node v split on the values of attribute $a \in A$ into $children(v,a)$ ...

Information Gain

$$InfGain(v,a) = entropy(v) - \sum_{u \in children(v,a)} \frac{|u|}{|v|} entropy(u)$$

Intrinsic Information

from Piatesky-Shapiro

$$IntrinsicInformation(v,a) = - \sum_{u \in children(v,a)} \frac{|u|}{|v|} \log \frac{|u|}{|v|}$$

Gain Ratio

from (Quinlan, 1986)

The gain ratio achieved when splitting v on a is

$$GainRatio(v,a) = \frac{InfGain(v,a)}{IntrinsicInformation(v,a)}$$

Tree Induction Algorithms - Information Gain and Gain Ratio

For a node v split on the values of attribute $a \in A$ into $children(v,a)$ ...

Information Gain

$$InfGain(v,a) = entropy(v) - \sum_{u \in children(v,a)} \frac{|u|}{|v|} entropy(u)$$

Intrinsic Information

from Piatesky-Shapiro

$$IntrinsicInformation(v,a) = -\sum_{u \in children(v,a)} \frac{|u|}{|v|} \log \frac{|u|}{|v|}$$

Gain Ratio

from (Quinlan, 1986)

The gain ratio achieved when splitting v on a is

$$GainRatio(v,a) = \frac{InfGain(v,a)}{IntrinsicInformation(v,a)}$$

Which attribute gives the best root split in the 'patient responses' dataset?

Tree Induction Algorithms - Binary Classifiers

Learning a binary classifier

INPUT: Training set D, Labelset $L = \{y_1, \ldots, y_k\}$, set of (attribute,value)-pairs $AV(A) \equiv AV$ for the set of attributes A

At the current node v, invoke
$binCore(v, L, AV)$

IF $\exists y \in L$ such that for most of the $x \in v : label(x) = y$ THEN do

set y as the label of the whole v and return

ELSE IF $AV = \emptyset$ THEN do

1. identify the majority class label of v, y
2. set y as the label of the whole v and return

ELSE do

1. Choose the best binary split of v into v_1, v_2
2. Remove from AV the pair that caused the best binary split
3. For each $u \in \{v_1, v_2\}$ invoke $binCore(u, L, AV)$

Progress and outlook

We have seen:

- √ How to build a decision tree by recursively splitting the dataset into more and more homogeneous nodes – where homogeneity refers to the target variable
- √ Split functions that implement different definitions of 'homogeneity'
- √ Different types of decision tree, depending on whether a node has exactly two children or as many children as the values of the splitting attribute

Each type of decision tree and each type of split function lead to a different classifier:

- How to figure out how good a classifier is?

Thank you very much! Questions?