

Week 8, Lecture 15 - Decision Trees

Aaron Meyer

Outline

- ▶ Administrative Issues
- ▶ Decision Trees
- ▶ Applications
- ▶ Implementation

Classification

- ▶ With MLR and PLSR we talked about supervised regression
 - ▶ Continuous Y and continuous predictions
- ▶ With clustering, we get discrete predictions
 - ▶ But no Y output
- ▶ In the next two lectures, we're going to talk about methods that give discrete Y predictions
 - ▶ Decision trees (today)
 - ▶ Support vector machines (next lecture)

Decision Tree Representation

- ▶ Classification of instances by sorting them down the tree from the root to some leaf node
 - ▶ **node** \approx test of some attribute
 - ▶ **branch** \approx one of the possible values for the attribute
- ▶ Decision trees represent a **disjunction of conjunctions of constraints on the attribute values of instances**
- ▶ Equivalent to a set of if-then-rules
 - ▶ each branch represents one if-then-rule
 - ▶ **if-part**: conjunctions of attribute tests on the nodes
 - ▶ **then-part**: classification of the branch

Decision Tree Representation



- This decision tree is equivalent to:
 - if (Outlook = Sunny) \wedge (Humidity = Normal) then Yes;
 - if (Outlook = Overcast) then Yes;

Appropriate Problems

- ▶ Instances are represented by attribute-value pairs, e.g. (Temperature, Hot)
- ▶ Target function has discrete output values, e.g. *yes* or *no*
- ▶ **Disjunctive descriptions** may be required
- ▶ Training data may **contain errors**
- ▶ Training data may contain **missing attribute values**
 - ▶ Last three points make Decision Tree Learning more attractive than CANDIDATE-ELIMINATION

ID3

- ▶ Learns decision trees by constructing them **top-down**
- ▶ employs a **greedy search algorithm without backtracking** through the space of all possible decision trees
 - ▶ finds the shortest but not necessarily the best decision tree
- ▶ **key idea:**
 - ▶ selection of the next attribute according to a statistical measure
 - ▶ all examples are considered at the same time (simultaneous covering)
 - ▶ recursive application with reduction of selectable attributes until each training example can be classified unambiguously

ID3 algorithm

ID3(Examples, Target_attribute, Attributes)

Create a Root for the tree

- ▶ If all examples are **positive**, Return single-node tree Root, with label = +
- ▶ If all examples are **negative**, Return single-node tree Root, with label = -
- ▶ If Attributes is empty, Return single-node tree Root, with label = most common value of Target_attribute in Examples

ID3 algorithm

ID3(Examples, Target_attribute, Attributes)

otherwise, Begin

- ▶ $A \leftarrow$ attribute in Attributes that best classifies Examples
decision attribute for Root $\leftarrow A$
- ▶ **For each possible value v_i of A**
 - ▶ Add new branch below Root with $A = v_i$
 - ▶ Let Examples_vi be the subset of Examples with v_i for A
 - ▶ If Examples is empty
 - ▶ Then add a leaf node with label = most common value of Target_attribute in Examples
 - ▶ Else add ID3(Examples_vi, Target_Attribute, Attributes - {A})
- ▶ Return Root

The best classifier

- ▶ **Central choice:** Which attribute classifies the examples best?
- ▶ ID3 uses the **information gain**
 - ▶ statistical measure that indicates how well a given attribute separates the training examples according to their target classification

$$Gain(S, A) = \underbrace{Entropy(S)}_{\text{original entropy of } S} - \underbrace{\sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)}_{\text{relative entropy of } S}$$

- ▶ Interpretation:
 - ▶ Denotes the reduction in entropy caused by partitioning S according to A
 - ▶ Alternative: number of saved yes/no questions (i.e., bits)
 - ▶ Attribute with $\max_A Gain(S, A)$ is selected!

Entropy

- ▶ Statistical measure from information theory that **characterizes (im-)purity** of an arbitrary collection of examples S
 - ▶ Definition: $H(S) \equiv \sum_{\forall i} -p_i \log_2 p_i$
- ▶ Interpretation
 - ▶ Specification of the minimum number of bits of information needed to encode the classification of an arbitrary member of S
 - ▶ Alternative: number of yes/no questions

Entropy



► Minimum of $H(S)$

Illustrative Example

Example days:

| Day | <i>Sunny</i> | <i>Temp.</i> | <i>Humidity</i> | <i>Wind</i> | <i>PlayTennis</i> |
|-----|--------------|--------------|-----------------|-------------|-------------------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

Illustrative Example

● entropy of S

$$S = \{D1, \dots, D14\} = [9+, 5-]$$

$$H(S) = -\frac{9}{14} \cdot \log_2 \frac{9}{14} - \frac{5}{14} \cdot \log_2 \frac{5}{14} = 0.940$$

● information gain (e.g. $Wind$)

$$S_{Weak} = \{D1, D3, D4, D5, D8, D9, D10, D13\} = [6+, 2-]$$

$$S_{Strong} = \{D2, D6, D7, D11, D12, D4\} = [3+, 3-]$$

$$\begin{aligned} Gain(S, Wind) &= H(S) - \sum_{v \in Wind} \frac{|S_v|}{|S|} \cdot H(S_v) \\ &= H(S) - \frac{8}{14} \cdot H(S_{Weak}) - \frac{6}{14} \cdot H(S_{Strong}) \\ &= 0.940 - \frac{8}{14} \cdot 0.811 - \frac{6}{14} \cdot 1.000 \\ &= 0.048 \end{aligned}$$

Illustrative Example

Which attribute is the best classifier?



$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= .940 - (7/14) \cdot 0.985 - (7/14) \cdot 0.592 \\ &= .151 \end{aligned}$$

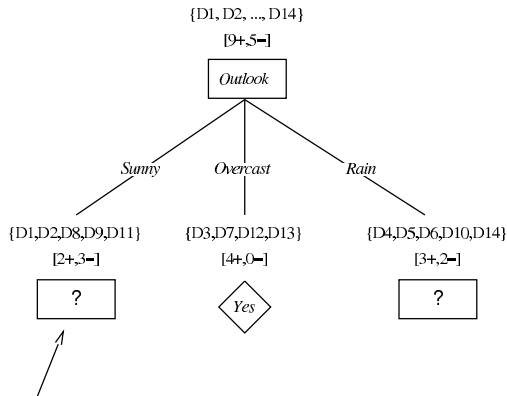


$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= .940 - (8/14) \cdot 0.811 - (6/14) \cdot 1.0 \\ &= .048 \end{aligned}$$

Illustrative Example

- ▶ Informations gains for the four attributes:
 - ▶ $Gain(S, Outlook) = 0.246$
 - ▶ $Gain(S, Humidity) = 0.151$
 - ▶ $Gain(S, Wind) = 0.048$
 - ▶ $Gain(S, Temperature) = 0.029$
- ▶ *Outlook* is selected as best classifier and is therefore root of the tree
- ▶ Now branches are created below the root for each possible value
 - ▶ Because every example for which $Outlook = Overcast$ is positive, this node becomes a leaf node with the classification *Yes*
 - ▶ The other descendants are still ambiguous (e.g. $H(S) \neq 0$)
 - ▶ Hence, the decision tree has to be further elaborated below these nodes

Illustrative Example



Which attribute should be tested here?

$$S_{\text{sunny}} = \{D1, D2, D8, D9, D11\}$$

$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

$$\text{Gain}(S_{\text{sunny}}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

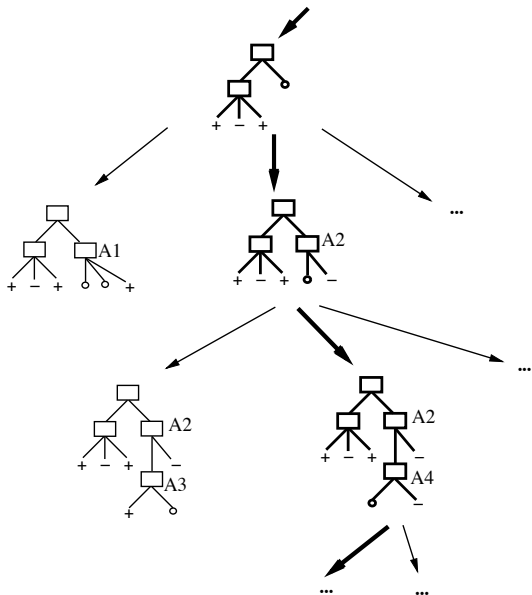
$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$

Illustrative Example

Resulting decision tree



Hypothesis Space Search



Inductive Bias

- ▶ As mentioned above, ID3 searches
 - ▶ Complete space of possible, but not completely
 - ▶ Preference Bias
- ▶ Inductive bias: Shorter trees are preferred to longer trees. Trees that place high information gain attributes close to the root are also preferred.
- ▶ Why prefer shorter hypotheses?
 - ▶ Occam's Razor: Prefer the simplest hypothesis that fits the data!
 - ▶ see Minimum Description Length Principle (Bayesian Learning)
 - ▶ e.g., if there are two decision trees, one with 500 nodes and another with 5 nodes, the second one should be preferred
 - ▶ better chance to avoid overfitting

Overfitting



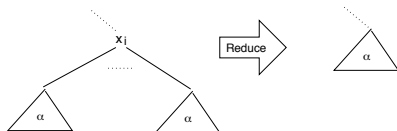
Given a hypothesis space H , a hypothesis $h \in H$ is said to overfit the training data if there exists some alternative hypothesis $h' \in H$, such that h has smaller error than h' over the training, but h' has

Overfitting

- ▶ Reasons for overfitting:
 - ▶ Noise in the data
 - ▶ Number of training examples is too small to produce a representative sample of the target function
- ▶ How to avoid overfitting:
 - ▶ Stop the tree growth earlier, before it reaches the point where it perfectly classifies the training data
 - ▶ Allow overfitting and then **post-prune** the tree (more successful in practice!)
- ▶ How to determine the perfect tree size:
 - ▶ Separate validation set to evaluate utility of post-pruning
 - ▶ Apply statistical test to estimate whether expanding (or pruning) produces an improvement

Reduced Error Pruning

- ▶ Each of the decision nodes is considered to be candidate for pruning



- ▶ Pruning a decision node consists of removing the subtree rooted at the node, making it a leaf node and assigning the most common classification of the training examples affiliated with that node
- ▶ Nodes are removed only if the resulting tree performs not worse than the original tree over the validation set
- ▶ Pruning starts with the node whose removal most increases accuracy and continues until further pruning is harmful

Reduced Error Pruning

Effect of reduced error pruning:



Any node added to coincidental regularities in the training set is likely to be pruned.

Rule Post-Pruning

- ▶ Rule post-pruning involves the following steps:
 1. Infer the decision tree from the training set (Overfitting allowed!)
 2. Convert the tree into a set of rules
 3. Prune each rule by removing any preconditions that result in improving its estimated accuracy
 4. Sort the pruned rules by their estimated accuracy
- ▶ One method to estimate rule accuracy is to use a separate validation set
- ▶ Pruning rules is more precise than pruning the tree itself

Alternative Measures

- ▶ Natural bias in information gain favors attributes with many values over those with few values
- ▶ e.g. attribute Date
 - ▶ Very large number of values (e.g. March 21, 2005)
 - ▶ Inserted in the above example, it would have the highest information gain, because it perfectly separates the training data
 - ▶ But the classification of unseen examples would be impossible
- ▶ Alternative measure: **GainRatio**
 - ▶ $GainRatio(S, A) = \frac{Gain(S, A)}{SplitInformation(S, A)}$
 - ▶ $SplitInformation(S, A) \equiv - \sum_{i=1}^n \frac{S_i}{S} \log_2 \frac{S_i}{S}$
 - ▶ $SplitInformation(S, A)$ is sensitive to how broadly and uniformly A splits S (entropy of S with respect to the values of A)
 - ▶ $GainRatio$ penalizes attributes such as Date

Summary

- ▶ Practical and intuitively understandable method for concept learning
- ▶ Able to learn disjunctive, discrete-valued concepts
- ▶ Noise in the data is allowed
- ▶ ID3 is a simultaneous covering algorithm based on information gain that performs a greedy top-down search through the space of possible decision trees
- ▶ Inductive Bias: Short trees are preferred (Occam's razor)
- ▶ Overfitting is an important issue and can be reduced by pruning

Predicting cesarean delivery with decision tree models

Cynthia J. Sims, MD,^a Leslie Meyn, BS,^a Rich Caruana, PhD,^{b, c} R. Bharat Rao, PhD,^d
Tom Mitchell, PhD,^b and Marijane Krohn, PhD^a

Pittsburgh, Pennsylvania, Los Angeles, California, and Princeton, New Jersey

OBJECTIVE: The purpose of this study was to determine whether decision tree-based methods can be used to predict cesarean delivery.

STUDY DESIGN: This was a historical cohort study of women delivered of live-born singleton neonates in 1995 through 1997 (22,157). The frequency of cesarean delivery was 17%; 78 variables were used for analysis. Decision tree rule-based methods and logistic regression models were each applied to the same 50% of the sample to develop the predictive training models and these models were tested on the remaining 50%.

RESULTS: Decision tree receiver operating characteristic curve areas were as follows: nulliparous, 0.82; parous, 0.93. Logistic receiver operating characteristic curve areas were as follows: nulliparous, 0.86; parous, 0.93. Decision tree methods and logistic regression methods used similar predictive variables; however, logistic methods required more variables and yielded less intelligible models. Among the 6 decision tree building methods tested, the strict minimum message length criterion yielded decision trees that were small yet accurate. Risk factor variables were identified in 676 nulliparous cesarean deliveries (69%) and 419 parous cesarean deliveries (47.6%).

CONCLUSION: Decision tree models can be used to predict cesarean delivery. Models built with strict minimum message length decision trees have the following attributes: Their performance is comparable to that of logistic regression; they are small enough to be intelligible to physicians; they reveal causal dependencies among variables not detected by logistic regression; they can handle missing values more easily than can logistic methods; they predict cesarean deliveries that lack a categorized risk factor variable. (Am J Obstet Gynecol 2000;183:1198-206.)

Key words: Decision trees, machine learning, predicting cesarean delivery, statistical models

Applications - Medical Diagnosis / Prediction

Table 1. Relationships between maternal and fetal characteristics and cesarean delivery

| Variable | Total | Cesarean delivery | | Statistical significance* |
|--|--------|-------------------|------|---------------------------|
| | | No. | % | |
| African American race | | | | $P < .001$ |
| No | 18,592 | 3221 | 17.3 | |
| Yes | 3,502 | 505 | 14.4 | |
| Missing | 63 | 9 | 14.3 | |
| Age | | | | $P < .001$ |
| 12-19 y | 1,797 | 165 | 9.2 | |
| 20-24 y | 3,222 | 460 | 14.3 | |
| 25-29 y | 5,949 | 886 | 14.9 | |
| 30-34 y | 7,062 | 1295 | 18.3 | |
| ≥ 35 y | 4,127 | 929 | 22.5 | |
| Married | | | | $P < .001$ |
| Yes | 14,641 | 2631 | 18.0 | |
| No | 7,516 | 1104 | 14.7 | |
| Insurance type | | | | $P < .001$ |
| Third-party | 16,847 | 2988 | 17.7 | |
| Public | 5,310 | 747 | 14.1 | |
| Previous cesarean delivery† | | | | $P < .001$ |
| No | 9,521 | 568 | 6.0 | |
| Yes | 2,439 | 1204 | 49.4 | |
| Cigarette use in pregnancy | | | | $P = .747$ |
| No | 19,329 | 3251 | 16.8 | |
| Yes | 2,819 | 481 | 17.1 | |
| Missing | 9 | 3 | 33.3 | |
| Diabetes | | | | $P < .001$ |
| None | 21,464 | 3508 | 16.3 | |
| Gestational | 552 | 157 | 28.4 | |
| Insulin-dependent | 137 | 67 | 48.9 | |
| Missing | 4 | 3 | 75.0 | |
| Hypertension | | | | $P < .001$ |
| None | 20,222 | 3217 | 15.9 | |
| Transient | 1,168 | 234 | 20.0 | |
| Preeclampsia or eclampsia | 767 | 284 | 37.0 | |
| Preterm delivery (< 37 wk) | | | | $P < .001$ |
| No | 19,790 | 3071 | 15.5 | |
| Yes | 2,365 | 664 | 28.1 | |
| Missing | 2 | 0 | 0 | |
| Fetal presentation | | | | $P < .001$ |
| Vertex | 20,550 | 2455 | 11.9 | |
| Breech | 855 | 740 | 86.5 | |
| Abnormal lie | 752 | 540 | 71.8 | |
| Fetal growth by <i>International Classification of Diseases, Ninth Revision</i> code | | | | $P < .001$ |
| Intrauterine growth restriction | 551 | 131 | 23.8 | |
| Normal | 20,167 | 3092 | 15.3 | |
| Macrosomia | 1,439 | 512 | 35.6 | |

*Missing cases were not included in calculations of P values.

†Among women who had at least one previous delivery.

Applications - Medical Diagnosis / Prediction



Applications - Medical Diagnosis / Prediction



Applications - Medical Diagnosis / Prediction

Table III. Variables used in decision tree and logistic regression nulliparous

| <i>Variable</i> | <i>Nulliparous</i> | |
|---------------------------------|----------------------|-----------------|
| | <i>Decision tree</i> | <i>Logistic</i> |
| African American race | | Used |
| Grouped age | Used | Used |
| History of cesarean delivery | | |
| Maternal admission weight | | Used |
| Diabetes | | Used |
| Structural heart disease | | Used |
| Hypertension | Used | Used |
| Other pulmonary condition | | |
| Placenta previa | | Used |
| Uterine abnormality | | Used |
| Herpes simplex virus infection | | Used |
| Preterm labor | | Used |
| Fetal growth | Used | Used |
| Preterm delivery | | Used |
| Abruptio placentae | Used | Used |
| Chorioamnionitis | Used | Used |
| Type of membrane rupture | | Used |
| Fetal presentation | Used | Used |
| Fetal distress | Used | Used |
| Umbilical cord prolapse | | |
| Meconium-stained amniotic fluid | | |

Practical Notes

- ▶ Building and applying decision trees is extremely fast.
- ▶ Results are very interpretable.
- ▶ With boosting, DTs often win at prediction with large data and many variables.
 - ▶ But this comes at the expense of interpreting the model.
- ▶ Bootstrap difficult to apply to DTs, due to structure.

Implementation

Further Reading

- ▶ Computer Age Statistical Inference, Chapter 8
- ▶ `sklearn.tree`