Lesson 10 Decision Tree

Lusine Zilfimian

April 27 (Monday), 2020

Contents

Contents

Contents

Contents

Contents

- Quiz
- How a Decision Tree Works
- How to Build a Decision Tree
- Attribute Test Conditions
- Impurity Measures

Contents

Last Lecture ReCap

- How to choose the value of k in k-NN classification?

Last Lecture ReCap

- How to choose the value of k in k-NN classification?
- What is the difference between k-NN classification and k-NN regression?

Last Lecture ReCap

- How to choose the value of k in k-NN classification?
- What is the difference between k-NN classification and k-NN regression?
- Could you solve the problem from the last lecture?

How a Decision Tree Works

- Suppose we want to predict the university where the student is studying.

How a Decision Tree Works

- Suppose we want to predict the university where the student is studying.
- Let we have 2 options: AUA and YSU (too boring, isn't it? $\ddot\smile$)

How a Decision Tree Works

- Suppose we want to predict the university where the student is studying.
- Let we have 2 options: AUA and YSU (too boring, isn't it? $\ddot\smile$)
- One approach is to ask a series of *questions* about the characteristics of a new student.

How a Decision Tree Works

- Suppose we want to predict the university where the student is studying.
- Let we have 2 options: AUA and YSU (too boring, isn't it? $\ddot\smile$)
- One approach is to ask a series of *questions* about the characteristics of a new student.
- Such...

How a Decision Tree Works

- Suppose we want to predict the university where the student is studying.
- Let we have 2 options: AUA and YSU (too boring, isn't it? $\ddot\smile$)
- One approach is to ask a series of *questions* about the characteristics of a new student.
- Such...
- The series of questions and answers can be represented as hierarchy of nodes and edges.

Nodes

- Root node

Nodes

- Root node
- Inernal node

Nodes

- Root node
- Inernal node
- Leaf or terminal node

Nodes

- Root node
- Inernal node
- Leaf or terminal node
- **Leaf node** is assigned a class label

How to Build a Decision Tree

- We can have different DTs from a sigle dataset.

How to Build a Decision Tree

- We can have different DTs from a sigle dataset.
- We will use a greedy strategy (locally optimum decision) to grow a decision tree.

How to Build a Decision Tree

- We can have different DTs from a sigle dataset.
- We will use a greedy strategy (locally optimum decision) to grow a decision tree.
- Hunt's Algorithm: partitioning the train data into purer subsets.

Example

- Suppose we have the following data

Table 1:

	Univ	DMGrade	Fail	MarStat
1	AUA	82	Yes	Single
2	AUA	67	No	Single
3	AUA	88	No	Divorced
4	AUA	72	No	Single
5	AUA	91	No	Married
6	YSU	76	Yes	Divorced
7	YSU	86	No	Divorced
8	YSU	87	Yes	Divorced
9	YSU	78	No	Single

Example

- Suppose we have the following data

Table 1:

	Univ	DMGrade	Fail	MarStat
1	AUA	82	Yes	Single
2	AUA	67	No	Single
3	AUA	88	No	Divorced
4	AUA	72	No	Single
5	AUA	91	No	Married
6	YSU	76	Yes	Divorced
7	YSU	86	No	Divorced
8	YSU	87	Yes	Divorced
9	YSU	78	No	Single

- Consider the problem of predicting the university of a student.

Example (Con'd)

- Hunt's algorithm will work if **every combination of attribute** values is present in the training data and each combination has a **unique class label**.

Example (Con'd)

- Hunt's algorithm will work if **every combination of attribute** values is present in the training data and each combination has a **unique class label**.
- Too stringent, isn't it?

Example (Con'd)

- Hunt's algorithm will work if **every combination of attribute** values is present in the training data and each combination has a **unique class label**.
- Too stringent, isn't it?
- Because some of the child nodes can be **empty** (assign majority vote of *parent* node)

Example (Con'd)

- Hunt's algorithm will work if **every combination of attribute** values is present in the training data and each combination has a **unique class label**.
- Too stringent, isn't it?
- Because some of the child nodes can be **empty** (assign majority vote of *parent* node)
- Or records can have **identical** attribute values (assign majority vote of *current* node)

The most interesting part

- How to split records?

The most interesting part

- How to split records?
- Select attribute

The most interesting part

- How to split records?
- Select attribute
- Select test condition

The most interesting part

- How to split records?
- Select attribute
- Select test condition
- Evaluate GoF

The most interesting part

- How to split records?
- Select attribute
- Select test condition
- Evaluate GoF
- How to stop splitting?

The most interesting part

- How to split records?
- Select attribute
- Select test condition
- Evaluate GoF
- How to stop splitting?
- All the records belong to the same class

The most interesting part

- How to split records?
- Select attribute
- Select test condition
- Evaluate GoF
- How to stop splitting?
- All the records belong to the same class
- Identical attribute values

The most interesting part

- How to split records?
- Select attribute
- Select test condition
- Evaluate GoF
- How to stop splitting?
- All the records belong to the same class
- Identical attribute values
- Terminate earlier (records have fallen below some minimum threshold)

Methods for Expressing Attribute Test Conditions

- Binary Attributes

Methods for Expressing Attribute Test Conditions

- Binary Attributes
- Categorical Attributes (CART, produce only binary splits)

Methods for Expressing Attribute Test Conditions

- Binary Attributes
- Categorical Attributes (CART, produce only binary splits)
- Continuous Attributes

Measures for Selecting the Best Split

- $p(i|t)$ - fraction of records belonging to class i at a given node t.

Measures for Selecting the Best Split

- $p(i|t)$ - fraction of records belonging to class i at a given node t.
- Calculate $p(YSU|1)$ for above example.

Measures for Selecting the Best Split

- $p(i|t)$ - fraction of records belonging to class i at a given node t.
- Calculate $p(YSU|1)$ for above example.
- Measures for Selecting the Best Splitare often based on the degree of impurity of the child nodes.

Measures for Selecting the Best Split

- $p(i|t)$ - fraction of records belonging to class i at a given node t.
- Calculate $p(YSU|1)$ for above example.
- Measures for Selecting the Best Splitare often based on the degree of impurity of the child nodes.
- Node with class distribution $(0, 1)$ has ...

Measures for Selecting the Best Split

- $p(i|t)$ - fraction of records belonging to class i at a given node t.
- Calculate $p(YSU|1)$ for above example.
- Measures for Selecting the Best Splitare often based on the degree of impurity of the child nodes.
- Node with class distribution $(0,1)$ has ...
- ... zero impurity

Measures for Selecting the Best Split

- $p(i|t)$ - fraction of records belonging to class i at a given node t.
- Calculate $p(YSU|1)$ for above example.
- Measures for Selecting the Best Splitare often based on the degree of impurity of the child nodes.
- Node with class distribution $(0, 1)$ has ...
- ... zero impurity
- Node with uniform class distribution $(0.5, 0.5)$ has ...

Measures for Selecting the Best Split

- $p(i|t)$ - fraction of records belonging to class i at a given node t.
- Calculate $p(YSU|1)$ for above example.
- Measures for Selecting the Best Splitare often based on the degree of impurity of the child nodes.
- Node with class distribution $(0, 1)$ has ...
- ... zero impurity
- Node with uniform class distribution $(0.5, 0.5)$ has ...
- ... the highest impurity

Impurity measures

- Gini

Impurity measures

- Gini
- Classification error

Impurity measures

- Gini
- Classification error
- Entropy

Impurity measures

- Gini
- Classification error
- Entropy
- χ^2

Impurity measures

- All three measures attain their maximum value when the class distribution is uniform

Impurity measures

- All three measures attain their maximum value when the class distribution is uniform
- Test condition may vary depending on the choice of impurity measure

Splitting of Attributes

- Choose between attribute A and B using weighted average of impurity measures

Splitting of Attributes

- Choose between attribute A and B using weighted average of impurity measures
- The smalles is preferrable

Splitting of Attributes

- Choose between attribute A and B using weighted average of impurity measures
- The smalles is preferrable
- At each successive stage, compare this measure across all possible splits in all variables

Splitting of Attributes

- Choose between attribute A and B using weighted average of impurity measures
- The smalles is preferrable
- At each successive stage, compare this measure across all possible splits in all variables
- Choose the split that reduces impurity the most

Splitting of Attributes

- Choose between attribute A and B using weighted average of impurity measures
- The smalles is preferrable
- At each successive stage, compare this measure across all possible splits in all variables
- Choose the split that reduces impurity the most
- Chosen split points become nodes on the tree

Splitting of Attributes

- Binary: compare two-way splits Gini indexes

Splitting of Attributes

- Binary: compare two-way splits Gini indexes
- Categorical: compare both two-way and multi-way splits Gini indexes

Splitting of Attributes

- Binary: compare two-way splits Gini indexes
- Categorical: compare both two-way and multi-way splits Gini indexes
- Continuous:

Splitting of Attributes

- Binary: compare two-way splits Gini indexes
- Categorical: compare both two-way and multi-way splits Gini indexes
- Continuous:
- Find splitting value

Splitting of Attributes

- Binary: compare two-way splits Gini indexes
- Categorical: compare both two-way and multi-way splits Gini indexes
- Continuous:
- Find splitting value
- Split positions are identified by taking the midpoints between two adjacent sorted values

Splitting of Attributes

- Binary: compare two-way splits Gini indexes
- Categorical: compare both two-way and multi-way splits Gini indexes
- Continuous:
- Find splitting value
- Split positions are identified by taking the midpoints between two adjacent sorted values
- Split positions located between two adjacent records with different class labels

Summary of DT

- Both for classification and regression

Summary of DT

- Both for classification and regression
- Both for decriptive and predictive analysis

Summary of DT

- Both for classification and regression
- Both for decriptive and predictive analysis
- Nonparametric approach

Summary of DT

- Both for classification and regression
- Both for decriptive and predictive analysis
- Nonparametric approach
- Computationally inexpensive

Summary of DT

- Both for classification and regression
- Both for decriptive and predictive analysis
- Nonparametric approach
- Computationally inexpensive
- Relatively easy to interpret

Summary of DT

- Both for classification and regression
- Both for decriptive and predictive analysis
- Nonparametric approach
- Computationally inexpensive
- Relatively easy to interpret
- Use top-down, recursive partitioning approach

Summary of DT

- Both for classification and regression
- Both for decriptive and predictive analysis
- Nonparametric approach
- Computationally inexpensive
- Relatively easy to interpret
- Use top-down, recursive partitioning approach
- Using **only a single** attribute at a time (decision boundaries are rectilinear)