Decision Trees for Regression

Lecture 16

Dr. Emmanuel Papadakis

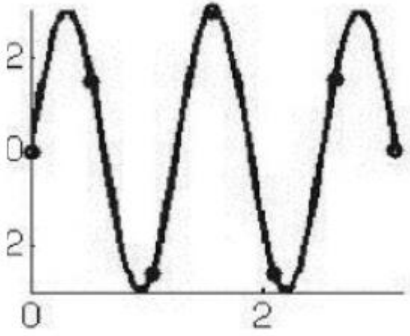
Overview

- Regression
- Decision Trees Classification Vs Regression
- Metrics for numerical purity
- ID3 algorithm for regression
- Example: use weather conditions to predict hours played

Regression

Recap of Regression

- **Problem:** Predict a numerical value **y**, given **x**
- **Strategy:** Use training samples (x,y) to find a mathematical function that approximates **f(x) = y**
- **Training process:** Fit a mathematical function to passes through the curve of training data as closely as possible

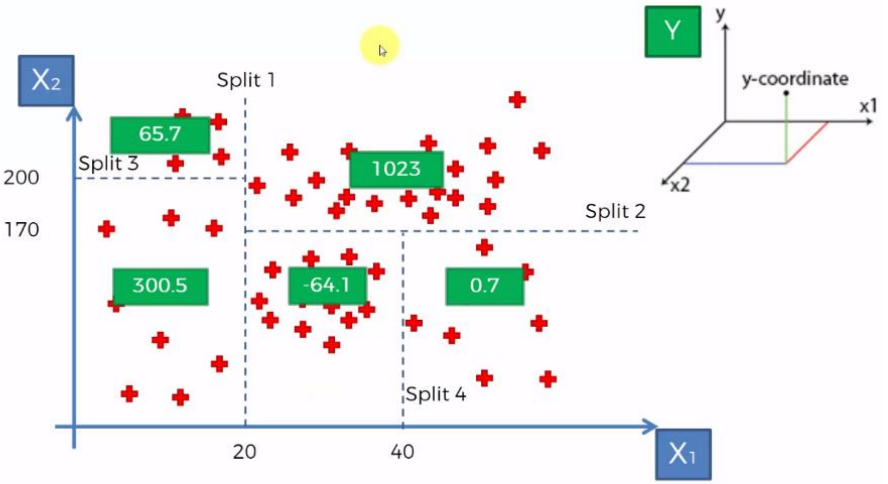


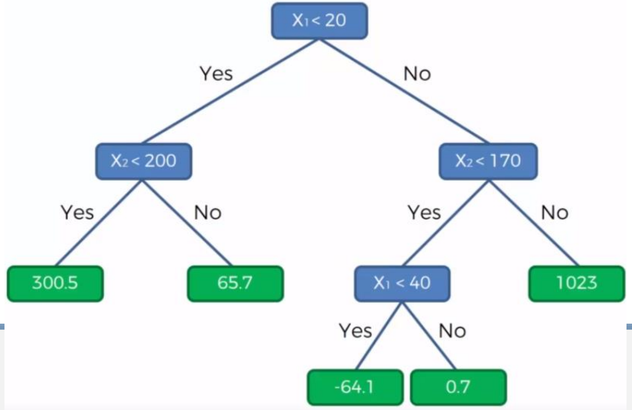
Overview

- Regression
- Decision Trees Classification Vs Regression
- Metrics for numerical purity
- ID3 algorithm for regression
- Example: use weather conditions to predict hours played

Classification Vs Regression Trees

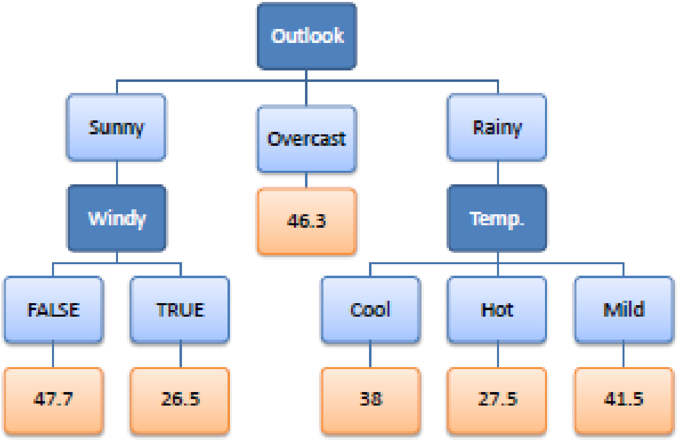
- Target feature is a continuous value
- Partition dataset into homogenuous subsets
- Conditions over the values of features create splits
- Decision nodes: values over the attribute tested
- Leaf nodes: predicted value





Regression Tree

Outlook	Temp.	Humidity	Windy	Hours Played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



Homogeneity during splits:

Classification – maximize Information Gain (reduce entropy of classified objects)

Regression – minimize Standard Deviation (reduce dispersion of target features)

Overview

- Regression
- Decision Trees Classification Vs Regression
- Metrics for numerical homogeneity
- ID3 algorithm for regression
- Example: use weather conditions to predict hours played

Metrics for numerical homogeneity

Hours Played
25
30
46
45
52
23
43
35
38
46
48
52
44
30

Count: n = 14

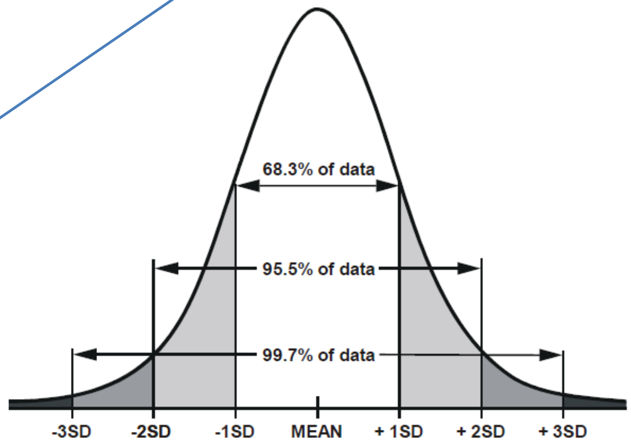
Mean: $\bar{x} = \frac{\sum x}{n} = 39.8$

Standard Deviation: $S = \sqrt{\frac{\sum(x-\bar{x})^2}{n}} = 9.32$

Amount of dispersion of the data set. If SD = 0, then the numerical sample is completely homogeneous.

Coefficient of Variation: $\mathrm{CV} = \frac{S}{\bar{x}} * 100\% = 23\%$

Ratio of dispersion around the mean.
Higher CV ➔ higher dispersion.



Standard deviation in a **normal distribution**

Measure SD of target feature after splitting with a predictor

predictor

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

target

- P(c) is the proportion of data instances, where predictor has a particular value C
- S(c) the standard deviation of the target feature when the predictor has a particular value C

What is the standard deviation for “Outlook”?

Outlook	Hours Played
Rainy	25
Rainy	30
Overcast	48
Sunny	45
Sunny	52
Sunny	23
Overcast	43
Rainy	35
Rainy	38
Sunny	48
Rainy	48
Overcast	52
Overcast	44
Sunny	30

Measure SD for each value of feature “Outlook”.

Overcast: {48, 43, 52, 44}
Rainy: {25, 30, 35, 38, 48}
Sunny: {45, 52, 23, 48, 30}

		Hours Played (SD)	Count	Mean
	Overcast	3.49	4	46.75
Outlook	Rainy	7.78	5	35.2
	Sunny	10.87	5	39.6
SUM			14	

$$S(Hours, Outlook)$$
$$= P(Overcast) * S(Overcast) + P(Rainy) * S(Rainy) + P(Sunny) * S(Sunny)$$
$$= \left(\frac{4}{14} * 3.49\right) + \left(\frac{5}{14} * 7.78\right) + \left(\frac{5}{14} * 10.78\right) = 7.66$$

Splitting tree with Standard Deviation Reduction

Measure the difference in data dispersion after a predictor splits the data into subsets.

- Pick the predictor that causes the highest reduction of dispersion

$$SDR(T,X) = S(T) - S(T,X)$$

- S(T) – Initial standard deviation of the target feature
- S(T, X) – standard deviation of predictor X

Apply Standard Deviation Reduction

S(Hours) = 9.32 ➔ initial data dispersion

Outlook	Hours Played
Rainy	25
Rainy	30
Overcast	48
Sunny	45
Sunny	52
Sunny	23
Overcast	43
Rainy	35
Rainy	38
Sunny	48
Rainy	48
Overcast	52
Overcast	44
Sunny	30

Standard Deviation Reduction
if data is split with "**Outlook**"

SDR(Hours, Outlook) =
S(Hours) – S(Hours, Outlook) =
9.32 – 7.66 = 1.66

S(Hours, Outlook) = 7.66 ➔ data dispersion after Outlook split

Standard Deviation Reduction

- Measure the effectiveness of a feature in reducing the dispersion of target values

Standard Deviation Reduction = **Reduction of Dispersion wrt Target/Prediction value**

Overview

- Regression
- Decision Trees Classification Vs Regression
- Metrics for numerical homogeneity
- ID3 algorithm for regression
- Example: use weather conditions to predict hours played

ID3 Algorithm for Regression

Build a decision tree, top-down from root

- Partition data into homogeneous subsets
- Use **Standard Deviation** to measure homogeneity after the dataset is split using a predictor (branching)
- Use **Coefficient of Variation** or **Count** to decide when to stop branching
  - Stop when the level of dispersion is acceptable or there are not enough observations
- Use **Mean** as the value of the leaf nodes

ID3 Algorithm for Regression: Pseudocode

ID3(Examples, Target, features)

Examples are the training examples S, Target is the target feature (the prediction)
features is the set of features maybe tested by the decision tree.
Return a decision tree that correctly predicts the target value of the given Examples.

Create a Root node for tree

If CV is less than the threshold return a single node tree with Mean

Otherwise Begin

A ←feature in features that best predicts S (with highest standard deviation reduction)

Set A as Root

for each possible value v of A

Add a new tree branch corresponding to A=v Let Sv
be the subset of examples in S with A=v
if Sv is empty: add leaf node with the average value of S (no observations)
Else: below this branch add a subtree

ID3(Sv, Label, features - {A})

End

Return Root

Overview

- Regression
- Decision Trees Classification Vs Regression
- Metrics for numerical homogeneity
- ID3 algorithm for regression
- Example: use weather conditions to predict hours played

Example: hours played based on weather conditions

Outlook	Temperature	Humidity	Windy	Hours Played
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Overcast	Hot	High	FALSE	48
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Overcast	Cool	Normal	TRUE	43
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Sunny	Mild	Normal	FALSE	48
Rainy	Mild	Normal	TRUE	48
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44
Sunny	Mild	High	TRUE	30

Step 1: Standard deviation of the entire population

S(Hours Played) = 9.32

Example: hours played based on weather conditions

Outlook	Temperature	Humidity	Windy	Hours Played
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Overcast	Hot	High	FALSE	48
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Overcast	Cool	Normal	TRUE	43
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Sunny	Mild	Normal	FALSE	48
Rainy	Mild	Normal	TRUE	48
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44
Sunny	Mild	High	TRUE	30

Step 2: Attempt splitting the dataset using different features.

Calculated the SDR for each feature

$$SDR(T,X) = S(T) - S(T,X)$$

$S(Hours, Outlook) = P(Overcast) * S(Overcast) + P(Rainy) * S(Rainy) + P(Sunny) * S(Sunny)$ = 7.66

SDR(Hours , Outlook) = **S**(Hours) – **S**(Hours, Outlook)

= 9.32 – 7.66 = 1.66

Select the feature with the highest reduction of dispersion

Example: hours played based on weather conditions

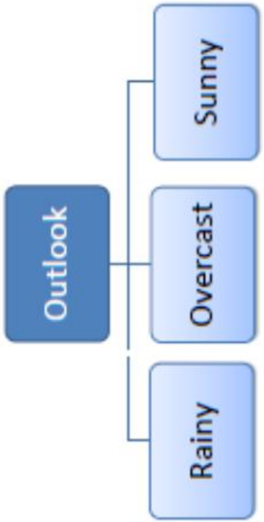
		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR= 0.48		

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

Example: hours played based on weather conditions



Outlook	Temp	Humidity	Windy	Hours Played
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Sunny	Mild	Normal	FALSE	46
Sunny	Mild	High	TRUE	30
Overcast	Hot	High	FALSE	46
Overcast	Cool	Normal	TRUE	43
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Rainy	Mild	Normal	TRUE	48

Step 3: Partition the data based on the best feature

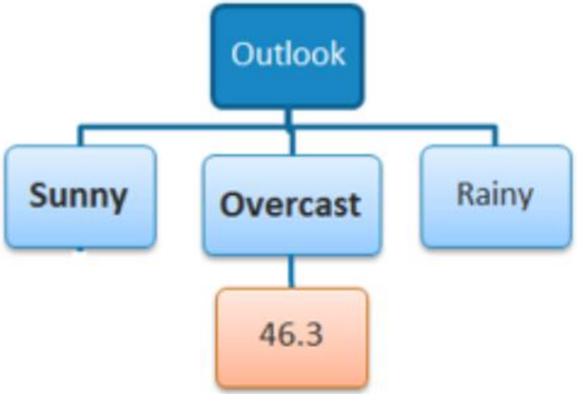
Continue **recursively** until

- Most of the data is processed (e.g., n > 3 to avoid overfitting)
- Target feature has an acceptable dispersion (e.g., CV < 10%)

Example: hours played based on weather conditions

Outlook - Overcast

		Hours Played (StDev)	Hours Played (AVG)	Hours Played (CV)	Count
Outlook	Overcast	3.49	46.3	8%	4
	Rainy	7.78	35.2	22%	5
	Sunny	10.87	39.2	28%	5



Overcast data subset does not need further splitting (CV < 10%).
A leaf node is generated with the Mean value of this sub-dataset.

Example: hours played based on weather conditions

Outlook - Sunny

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Cool	Normal	TRUE	23
Mild	Normal	FALSE	46
Mild	High	TRUE	30
			S = 10.87
			AVG = 39.2
			CV = 28%

		Hours Played (StDev)	Count
Temp	Cool	14.50	2
	Mild	7.32	3

SDR = 10.87-((2/5)*14.5 + (3/5)*7.32) = **0.678**

		Hours Played (StDev)	Count
Humidity	High	7.50	2
	Normal	12.50	3

SDR = 10.87-((2/5)*7.5 + (3/5)*12.5) = **0.370**

		Hours Played (StDev)	Count
Windy	False	3.09	3
	True	3.50	2

SDR = 10.87-((3/5)*3.09 + (2/5)*3.5) = **7.62**

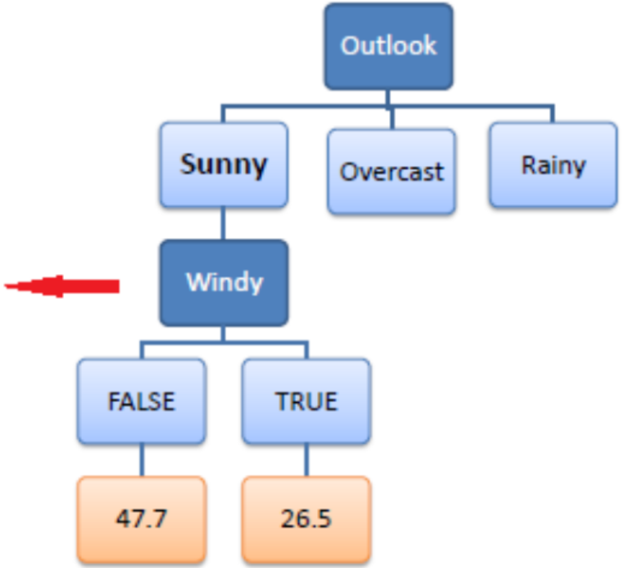
CV is not acceptable (28%).

Continue branching:
Use the new sub-data set.

“Windy“ partitions the dataset and achieves the lowest dispersion.

Example: hours played based on weather conditions

Temp.	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Mild	Normal	FALSE	46
Cool	Normal	TRUE	23
Mild	High	TRUE	30



Two branches are generated after splitting with “Windy”.

Both branches have 3 or less observations. (**terminating criterion**)

The algorithm stops and the mean values are assigned as leaf nodes.

Example: hours played based on weather conditions

Outlook - Rainy

Temp	Humidity	Windy	Hours Played
Hot	High	FALSE	25
Hot	High	TRUE	30
Mild	High	FALSE	35
Cool	Normal	FALSE	38
Mild	Normal	TRUE	48
			S = 7.78
			AVG = 35.2
			CV = 22%

		Hours Played (StDev)	Count
Temp	Cool	0	1
	Hot	2.5	2
	Mild	6.5	2

SDR = 7.78 - ((1/5)*0+(2/5)*2.5 + (2/5)*6.5) = 4.18

		Hours Played (StDev)	Count
Humidity	High	4.1	3
	Normal	5.0	2

SDR = 7.78 - ((3/5)*4.1 + (2/5)*5.0) = 3.32

		Hours Played (StDev)	Count
Windy	False	5.6	3
	True	9.0	2

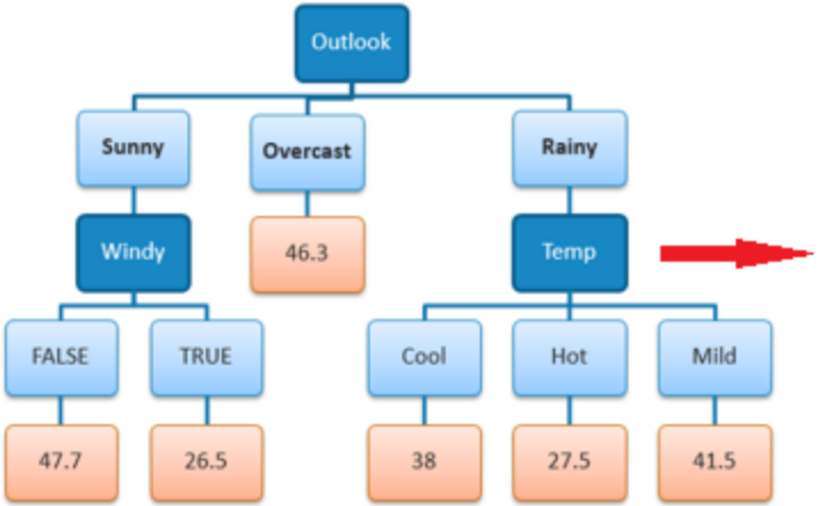
SDR = 7.78 - ((3/5)*5.6 + (2/5)*9.0) = 0.82

CV is not acceptable (22%).

Continue branching:
Use the new sub-dataset.

“Temperature“ partitions the dataset and achieves the highest SDR.

Example: hours played based on weather conditions



Temp	Hours Played
Cool	38
Hot	25
Hot	30
Mild	35
Mild	48

The remaining data instances are less than 3 for each branch generated by "Temperature" split.

➔ Stop branching

Generate leaf nodes for each branch with value the average of the assigned instances.

Summary of the ID3 algorithm

University of HUDDERSFIELD Inspiring global professionals

- ID3 conducts greedy search through space of possible decision nodes using **SDR** as heuristics

- grows the tree top-down, at each node selecting the feature with the largest reduction of data variation that best predicts the local training examples

- Stopping criteria
  - Every feature has been used along a specific path
  - The level of dispersion is within an acceptable threshold
  - There are not enough data instances to create a decision node

- Bonus: What happens when predictors are numerical?
  - Partition their values into chunks using thresholds. (e.g., 10 step process)