Lecture 8: 29 April, 2021

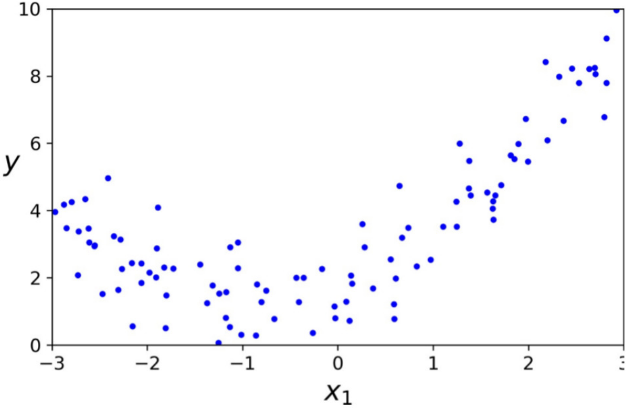
Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
April–July 2021

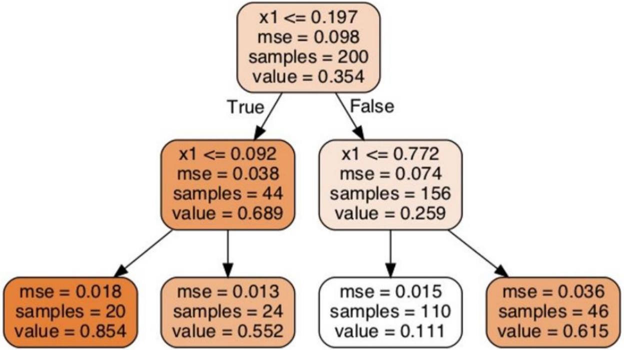
Decision trees for regression

- How do we use decision trees for regression?
- Partition the input into intervals
- For each interval, predict mean value of output, instead of majority class
- Regression tree

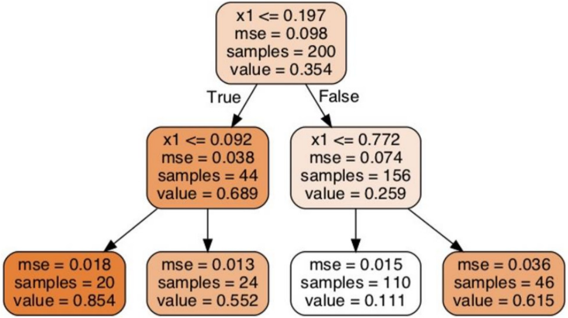


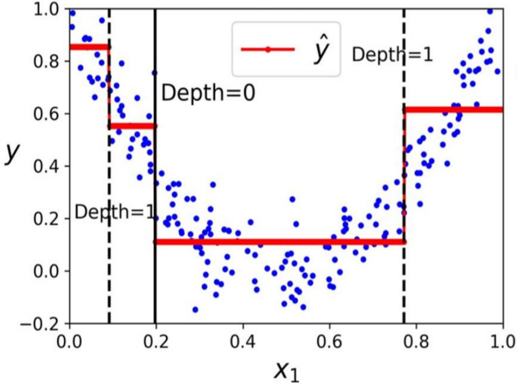
Decision trees for regression

- Regression tree for noisy quadratic centered around $x_1 = 0.5$
- For each node, the output is the mean y value for the current set of points
- Instead of impurity, use mean squared error (MSE) as cost function
- Choose a split that minimizes MSE



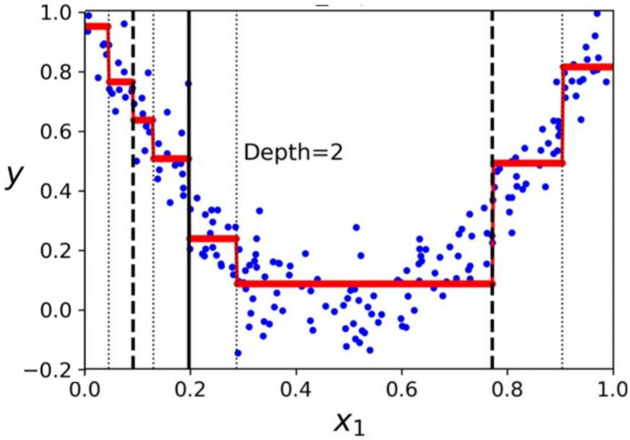
- Approximation using regression tree





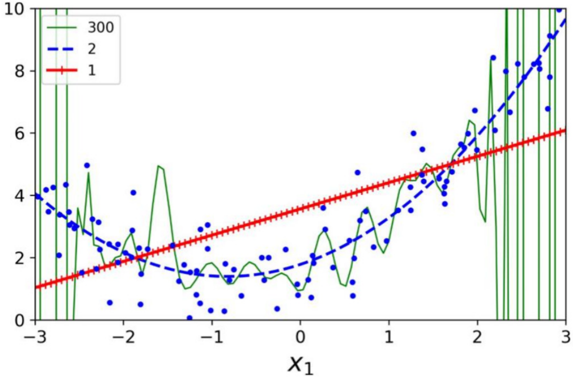
Regression trees

- Extend the regression tree one more level to get a finer approximation
- Set a threshold on MSE to decide when to stop
- Classification and Regression Trees (CART)
  - Combined algorithm for both use cases
- Programming libraries typically provide CART implementation



Overfitting

- Overfitting: model too specific to training data, does not generalize well
- Regression — use regularization to penalize model complexity
- What about decision trees?
- Deep, complex trees ask too many questions
- Prefer shallow, simple trees

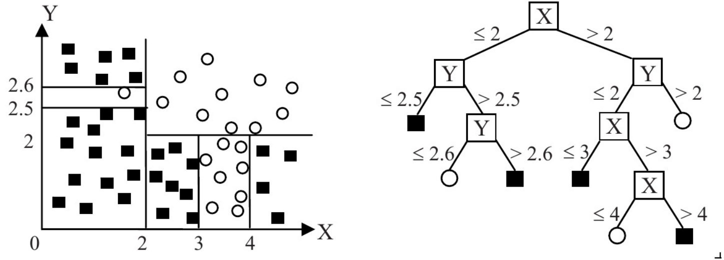


Tree pruning

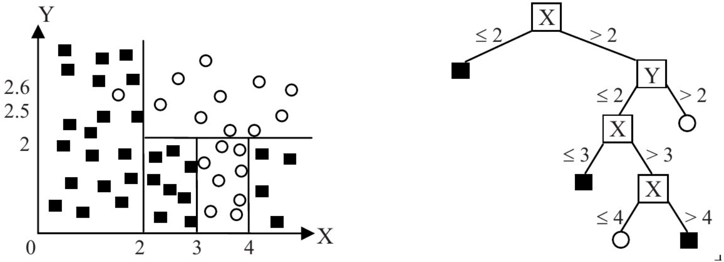
- Remove leaves to improve generalization
- Top-down pruning
  - Fix a maximum depth when building the tree
  - How to decide the depth in advance?
- Bottom-up pruning
  - Build the full tree
  - Remove a leaf if the reduced tree generalizes better
  - How do we measure this?

Tree pruning

Overfitted tree



Pruned tree



Bottom up tree pruning

- Build the full tree, remove leaf if the reduced tree generalizes better
- How do we measure this?
- Check performance on a test set
- Use sampling theory [Quinlan]
- Given n coin tosses with h heads, estimate probability of heads as h/n
  - Estimate comes with a confidence interval: $h/n \pm \delta$
  - As n increases, δ reduces: 7 heads out of 10 vs 70 out of 100 vs 700 out of 1000
- Impure node, majority prediction, compute confidence interval
- Pruning leaves creates a larger impure sample one level above
- Does the confidence interval decrease (improve)?

Example: Predict party from voting pattern [Quinlan]

- Predict party affiliation of US legislators based on voting pattern
  - Read the tree from left to right
- After pruning, drastically simplified tree
- Quinlan's comment on his use of sampling theory for post-pruning

Now, this description does violence to statistical notions of sampling and confidence limits, so the reasoning should be taken with a large grain of salt. Like many heuristics with questionable underpinnings, however, the estimates it produces seem frequently to yield acceptable results.

```
physician fee freeze = n:
|   adoption of the budget resolution = y: democrat (151)
|   adoption of the budget resolution = u: democrat (1)
|   adoption of the budget resolution = n:
|   |   education spending = n: democrat (6)
|   |   education spending = y: democrat (9)
|   |   education spending = u: republican (1)
physician fee freeze = y:
|   synfuels corporation cutback = n: republican (97/3)
|   synfuels corporation cutback = u: republican (4)
|   synfuels corporation cutback = y:
|   |   duty free exports = y: democrat (2)
|   |   duty free exports = u: republican (1)
|   |   duty free exports = n:
|   |   |   education spending = n: democrat (5/2)
|   |   |   education spending = y: republican (13/2)
|   |   |   education spending = u: democrat (1)
physician fee freeze = u:
|   water project cost sharing = n: democrat (0)
|   water project cost sharing = y: democrat (4)
|   water project cost sharing = u:
|   |   mx missile = n: republican (0)
|   |   mx missile = y: democrat (3/1)
|   |   mx missile = u: republican (2)
```

Bayesian classifiers

- As before
  - Attributes $\{A_1, A_2, \ldots, A_k\}$ and
  - Classes $C = \{c_1, c_2, \ldots c_\ell\}$
- Each class c_i defines a probabilistic model for attributes
  - $Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i)$
- Given a data item $d = (a_1, a_2, \ldots, a_k)$, identify the best class c for d
- Maximize $Pr(C = c_i \mid A_1 = a_1, \ldots, A_k = a_k)$

Generative models

- To use probabilities, need to describe how data is randomly generated
  - Generative model
- Typically, assume a random instance is created as follows
  - Choose a class c_j with probability $Pr(c_j)$
  - Choose attributes $a_1, \ldots, a_k$ with probability $Pr(a_1, \ldots, a_k \mid c_j)$
- Generative model has associated parameters $\theta = (\theta_1, \ldots, \theta_m)$
  - Each class probability $Pr(c_j)$ is a parameter
  - Each conditional probability $Pr(a_1, \ldots, a_k \mid c_j)$ is a parameter
- We need to estimate these parameters

Maximum Likelihood Estimators

- Our goal is to estimate parameters (probabilities) $\theta = (\theta_1, \ldots, \theta_m)$
- Law of large numbers allows us to estimate probabilities by counting frequencies
- Example: Tossing a biased coin, single parameter $\theta = Pr(\text{heads})$
  - N coin tosses, H heads and T tails
  - Why is $\hat{\theta} = H/N$ the best estimate?
- Likelihood
  - Actual coin toss sequence is $\tau = t_1 t_2 \ldots t_N$
  - Given an estimate of θ, compute $Pr(\tau \mid \theta)$ — likelihood $L(\theta)$
- $\hat{\theta} = H/N$ maximizes this likelihood — $\arg\max_{\theta} L(\theta) = \hat{\theta} = H/N$
  - Maximum Likelihood Estimator (MLE)

Bayesian classification

- Maximize $Pr(C = c_i \mid A_1 = a_1, \ldots, A_k = a_k)$
- By Bayes' rule,

$$Pr(C = c_i \mid A_1 = a_1, \ldots, A_k = a_k)$$

$$= \frac{Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i) \cdot Pr(C = c_i)}{Pr(A_1 = a_1, \ldots, A_k = a_k)}$$

$$= \frac{Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i) \cdot Pr(C = c_i)}{\sum_{j=1}^{\ell} Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_j) \cdot Pr(C = c_j)}$$

- Denominator is the same for all c_i, so sufficient to maximize

$$Pr(A_1 = a_1, \ldots, A_k = a_k \mid C = c_i) \cdot Pr(C = c_i)$$

Example

- To classify $A = g, B = q$

- $Pr(C = t) = 5/10 = 1/2$
- $Pr(A = g, B = q \mid C = t) = 2/5$
- $Pr(A = g, B = q \mid C = t) \cdot Pr(C = t) = 1/5$

- $Pr(C = f) = 5/10 = 1/2$
- $Pr(A = g, B = q \mid C = f) = 1/5$
- $Pr(A = g, B = q \mid C = f) \cdot Pr(C = f) = 1/10$

- Hence, predict $C = t$

A	B	C
m	b	t
m	s	t
g	q	t
h	s	t
g	q	t
g	q	f
g	s	f
h	b	f
h	q	f
m	b	f

Example . . .

- What if we want to classify $A = m, B = q$?
- $Pr(A = m, B = q \mid C = t) = 0$
- Also $Pr(A = m, B = q \mid C = f) = 0$!

A	B	C
m	b	t
m	s	t
g	q	t
h	s	t
g	q	t
g	q	f
g	s	f
h	b	f
h	q	f
m	b	f