CSE 446: Decision Trees

Winter 2012

Slides adapted from Carlos Guestrin and Andrew Moore by Luke Zettlemoyer & Dan Weld

Logistics

- Office Hours – see website
- Change in dataset
- Learning DTs with Attributes that have
  - Numerous Possible Values
  - Continuous Values
  - Missing Values
- Loss Functions

2

Overview of Learning

Type of Supervision
(eg, Experience, Feedback)

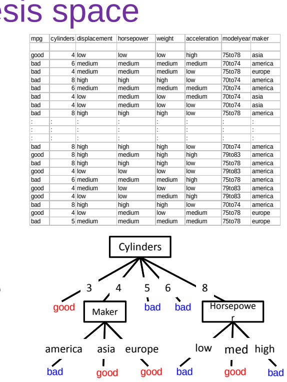
What is Being Learned?

	Labeled Examples	Reward	Nothing
Discrete Function	Classification		Clustering
Continuous Function	Regression		
Policy	Apprenticeship Learning	Reinforcement Learning	

5

Hypothesis space

- How many possible hypotheses?
- What functions can be represented?
- How many will be consistent with a given dataset?
- How will we choose the best one?

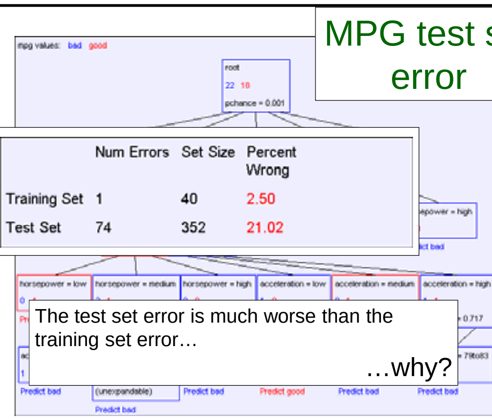


Two Questions

Greedy Algorithm:
- Start from empty decision tree
- Split on the **best attribute (feature)**
- Recurse

1. Which attribute gives the best split?
2. When to stop recursion?

MPG test set error



	Num Errors	Set Size	Percent Wrong
Training Set	1	40	2.50
Test Set	74	352	21.02

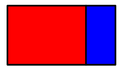
The test set error is much worse than the training set error…

…why?

1

Reduced Error Pruning

Split data into ***training*** & ***validation*** sets (10-33%)



Train on training set (overfitting)

Do until further pruning is harmful:

1) Evaluate effect on validation set of pruning ***each*** possible node (and tree below it)
2) Greedily remove the node that ***most improves accuracy of validation set***

A chi-square test

- Suppose that mpg was completely *uncorrelated* with maker.
- What is the chance we'd have seen data of at least this apparent level of association anyway?

By using a particular kind of chi-square test, the answer is 13.5%

Such hypothesis tests are relatively easy to compute, but involved

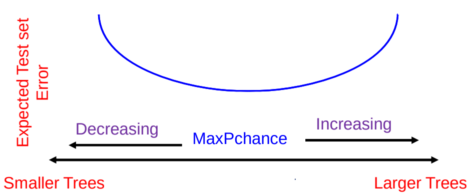
Using Chi-squared to avoid overfitting

- Build the full decision tree as before
- But when you can grow it no more, start to prune:
  - Beginning at the bottom of the tree, delete splits in which p_{chance} > *MaxPchance*
  - Continue working you way up until there are no more prunable nodes

MaxPchance is a magic parameter you must specify to the decision tree, indicating your willingness to risk fitting noise

Regularization

- Note for Future: MaxPchance is a regularization parameter that helps us bias towards simpler models



We'll learn to choose the value of magic parameters like this one later!

ML as Optimization

- Greedy search for best ***scoring*** hypothesis
- Where ***score*** =
  - Fits training data most accurately?
  - Sum: ***training accuracy – complexity penalty***

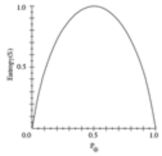
regularization

To be continued…

Advanced Decision Trees

- Attributes with:
  - Numerous Possible Values
  - Continuous (Ordered) Values
  - Missing Values

Information Gain



IG of attribute =

Decrease in entropy (uncertainty) after splitting

$$IG(X) = H(Y) - H(Y \mid X)$$

$$H(Y \mid X) = -\sum_{j=1}^{v} P(X=x_j) \sum_{i=1}^{k} P(Y=y_i \mid X=x_j) \log_2 P(Y=y_i \mid X=x_j)$$

Many Attribute Values

- What if split on S/N?
- H(MPG | S/N) ?

S/N	MPG
109788	good
245881	bad
7611005	good
299733	bad
445190	good
141178	bad
650998	good
120743	bad
252880	good
275003	bad
882301	good

$$H(Y \mid X) = -\sum_{j=1}^{v} P(X=x_j) \sum_{i=1}^{k} P(Y=y_i \mid X=x_j) \log_2 P(Y=y_i \mid X=x_j)$$

Attributes with many values

- **So many values that splitting on attribute…**
  - Divides examples into tiny sets
  - Each set likely homogeneous → high info gain
  - But poor predictor…
- **Need to penalize these attributes**
  - S/N is worst case, but correction is often needed

One Approach: Gain Ratio

$$GainRatio(S,A) \equiv \frac{\text{IG}\ (S,A)}{SplitInformation(S,A)}$$

$$SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where S_i is subset of S for which A has value v_i

SplitInfo ≅ entropy of S *wrt* values of A

(Contrast with entropy of S *wrt* **target** value)

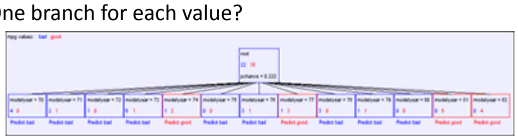
⇓ Attributes with many evenly distributed values

SplitInformation = $\log_2(n)$… = 1 for Boolean

Ordinal (Real-Valued) Inputs

What should we do if some of the inputs are real-valued?

One branch for each value?



Hopeless overfitting

Good News: GainRatio will reject these!

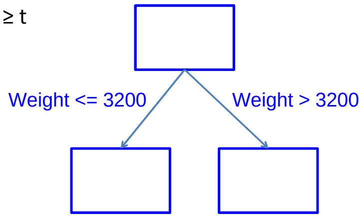
Bad News: They might have useful info!

Eg, weight of car might ≈ mpg

So…?

Threshold Splits

- Binary tree, split on attribute X at value t
  - One branch: X < t
  - Other branch: X ≥ t



The set of possible thresholds

- Binary tree, split on attribute X
  - One branch: X < t
  - Other branch: X ≥ t
- Search through possible values of *t*
  - Seems hard!!!
- But only finite number of *t*'s are important
  - Sort data according to X into $\{x_1,\ldots,x_m\}$
  - Consider split points of the form $x_i + (x_{i+1} - x_i)/2$
- Consider all such splits?

Only Certain Splits Make Sense

weight	MPG
2020	good
2600	good
3100	good
3500	bad
4200	good
4400	bad
4600	bad
6000	bad
7200	good
7800	bad
7995	bad

← ?

Picking the best threshold

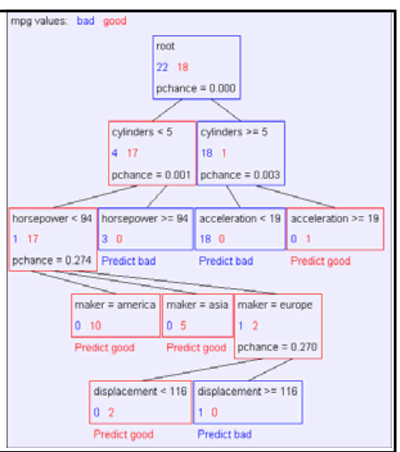
- Suppose X is real valued
- Define *IG(Y|X:t)* as *H(Y) - H(Y|X:t)*
- Define *H(Y|X:t) =*

  $H(Y|X < t)\ P(X < t) + H(Y|X >= t)\ P(X >= t)$

  - *IG(Y|X:t)* is the information gain for predicting Y if all you know is whether X is greater than or less than *t*
- Then define $IG^*(Y|X) = \max_t IG(Y|X:t)$
- For each real-valued attribute, use *IG*(Y|X)* for assessing its suitability as a split
- Note, may split on an attribute multiple times, with different thresholds

Example Tree for MPG

treating values as ordinal



Missing Data

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	missing	medium	70to74	america
bad	4	medium	medium	missing	low	75to78	europe
bad	8	high	high	missing	low	70to74	america
bad	6	medium	medium	missing	medium	70to74	america
bad	4	low	medium	missing	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	missing	low	75to78	america
good	4	low	low	missing	low	79to83	america
bad	6	medium	medium	missing	high	75to78	america
good	4	medium	low	missing	low	79to83	america
good	4	low	low	missing	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe

What to do??

Options

mpg	cylinders	displacement	horsepower	weight	ac
good	4	low	low	low	hig
bad	6	medium	medium	missing	me
bad	4	medium	medium	missing	low
bad	8	high	high	missing	low
bad	6	medium	medium	missing	me
bad	4	low	medium	missing	me
bad	4	low	medium	missing	low
bad	8	high	high	high	low

- Use “missing” as value
- Don’t use attribute at all
- Assign most common value to missing attribute
- Fractional values

Fractional Values

mpg	cylinders	displacement	horsepower	weight	ac
good	4	low	low	low	hig
bad	6	medium	medium	missing	me
bad	4	medium	medium	missing	low
bad	8	high	high	missing	low
bad	6	medium	medium	missing	me
bad	4	low	medium	missing	me
bad	4	low	medium	low	low
bad	8	high	high	high	low

- 66% low   and 33% high
- Training
  - Use in gain calculations
  - Further subdivide if other missing attributes
- Test
  - Classification is most ***probable classification***
  - Averaging over leaves where it got divided

What you need to know about decision trees

- Decision trees are one of the most popular ML tools
  - Easy to understand, implement, and use
  - Computationally cheap (to solve heuristically)
- Information gain to select attributes (ID3, C4.5,…)
- Presented for classification, can be used for regression and density estimation too
- Decision trees will overfit!!!
  - Must use tricks to find “simple trees”, e.g.,
    - Fixed depth/Early stopping
    - Pruning
    - Hypothesis testing

Loss Functions

- How measure quality of hypothesis?

Loss Functions

How measure quality of hypothesis?

L(x, y, $\hat{y}$) = utility(result of using y given input of x)
- utility(result of using $\hat{y}$ given input of x)

L(edible, poison)

L(poison, edible)

Common Loss Functions

- 0/1 loss 0 if y=$\hat{y}$ else 1
- Absolute value loss $|y-\hat{y}|$
- Squared error loss $|y-\hat{y}|^2$

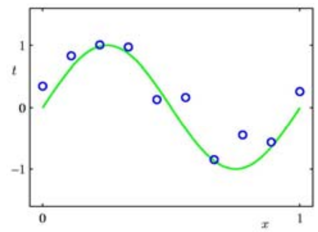
Overview of Learning

Type of Supervision
(eg, Experience, Feedback)

What is Being Learned?

	Labeled Examples	Reward	Nothing
Discrete Function	Classification		Clustering
Continuous Function	Regression		
Policy	Apprenticeship Learning	Reinforcement Learning	

Polynomial Curve Fitting

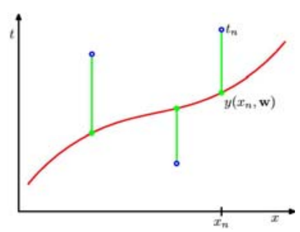


Sin(2πx)

Hypothesis Space

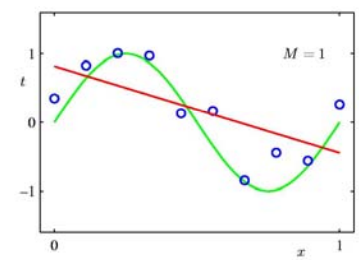
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sum-of-Squares Error Function

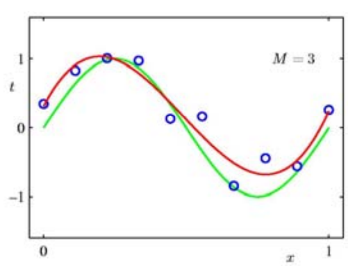


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

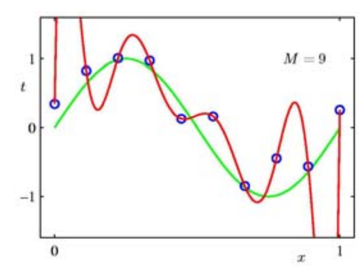
1st Order Polynomial



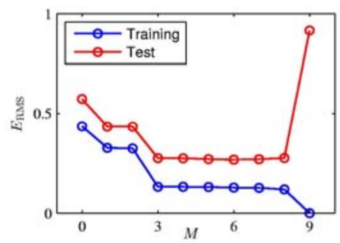
3rd Order Polynomial



9th Order Polynomial



Over-fitting



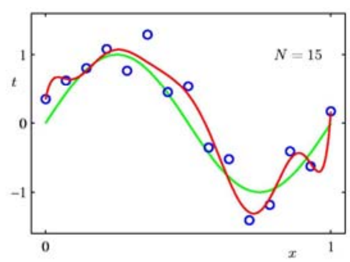
Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

Polynomial Coefficients

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

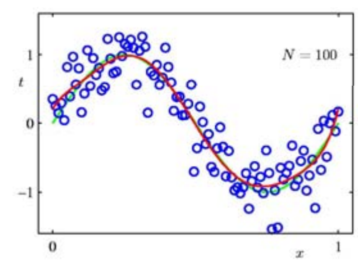
Data Set Size: $N = 15$

9th Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial

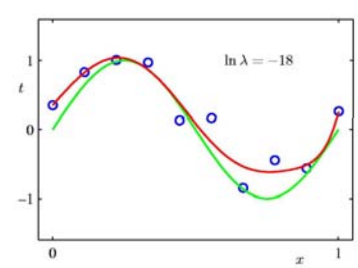


Regularization

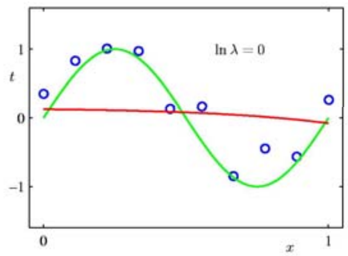
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Penalize large coefficient values

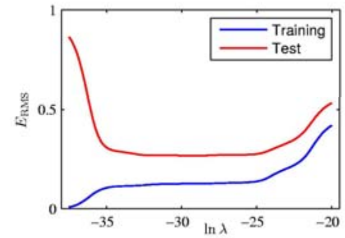
Regularization: $\ln\lambda = -18$



Regularization: $\ln \lambda = 0$



Regularization: E_{RMS} *vs.* $\ln \lambda$



Polynomial Coefficients

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Acknowledgements

- Some of the material in the decision trees presentation is courtesy of Andrew Moore, from his excellent collection of ML tutorials:
  - http://www.cs.cmu.edu/~awm/tutorials
- Improved by
  - Carlos Guestrin &
  - Luke Zettlemoyer