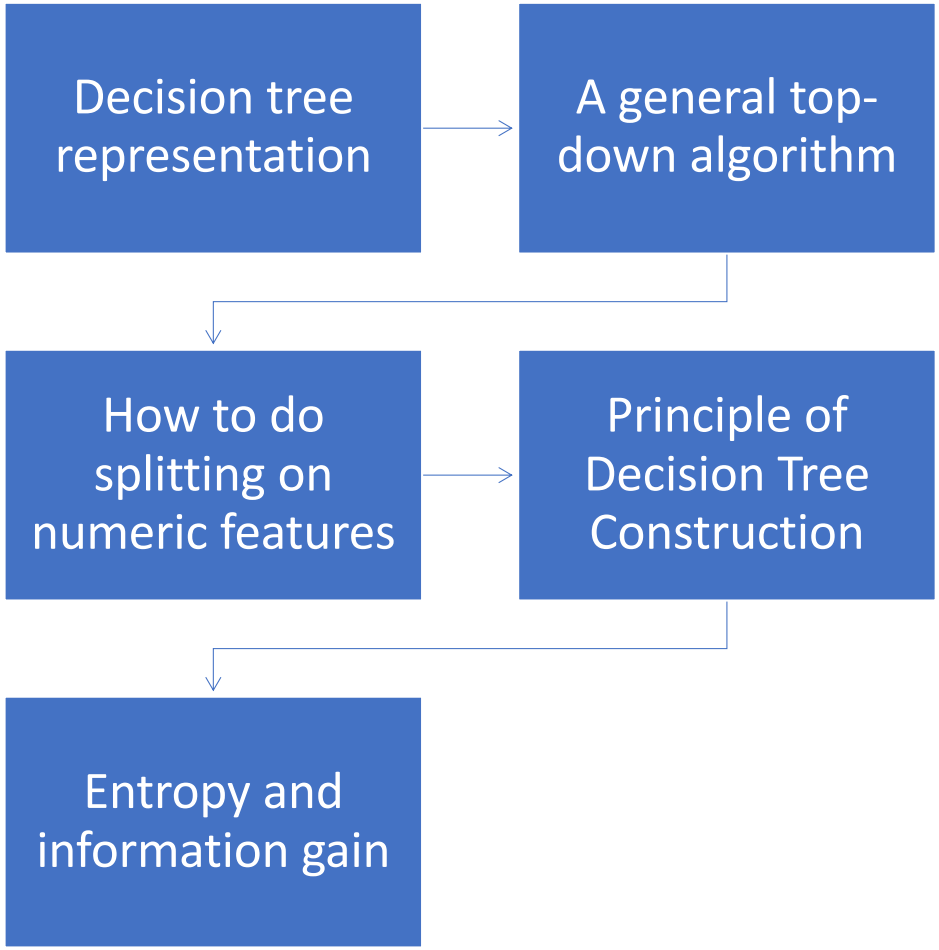
Lecture 9 -- Overfitting

Prof. Xiaowei Huang

https://cgi.csc.liv.ac.uk/~xiaowei/

(Attendance Code: **488324**)

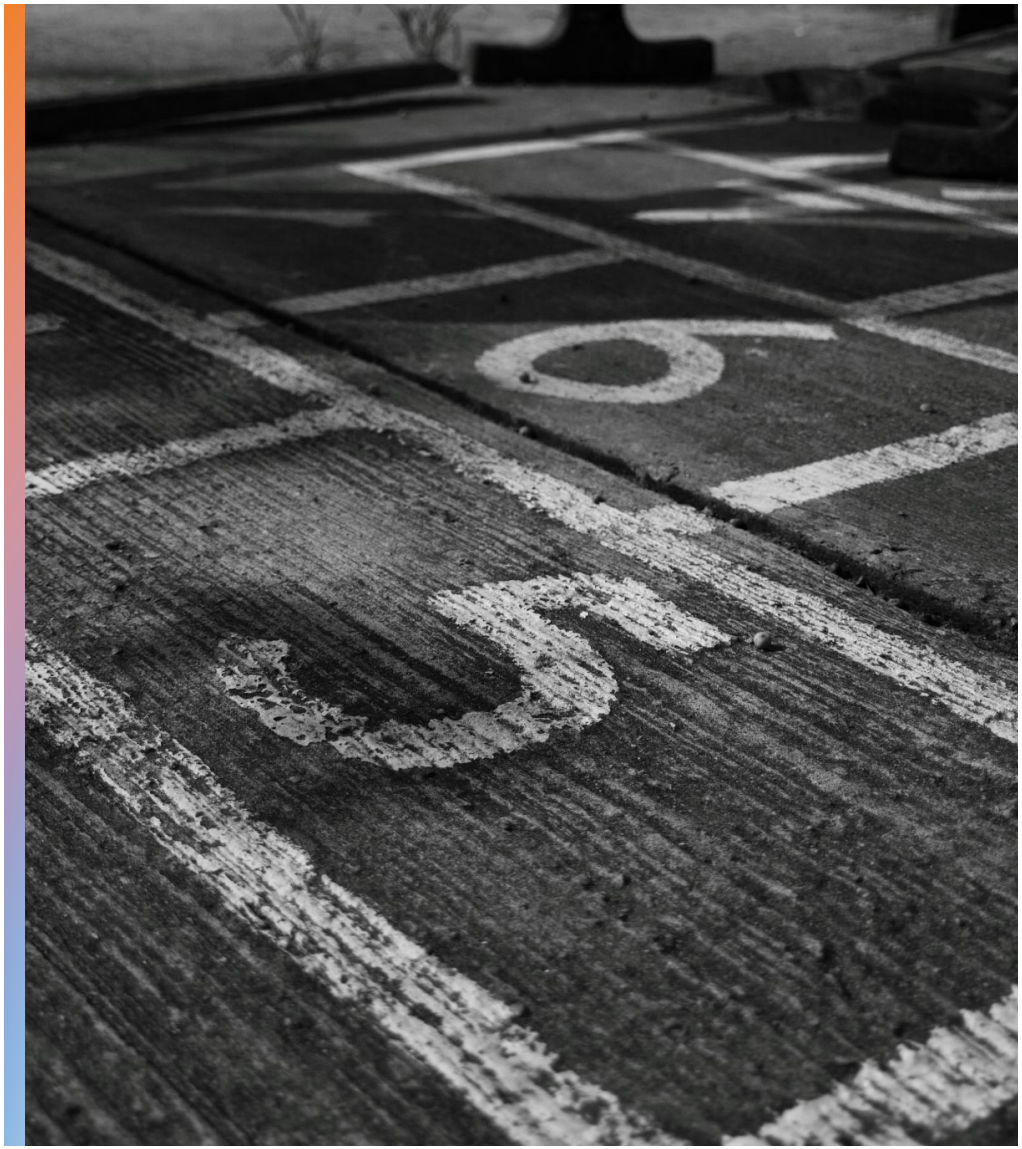
Decision Tree up to now,



Today's Topics

1. Stopping criteria of decision trees
2. Accuracy of decision trees
3. Pruning and validation dataset

4. Overfitting
5. Overfitting in decision tree



1. Step (3): Stopping criteria

Stopping criteria

- We should form a leaf when
  - all of the given subset of instances are of the same class, or
  - we've exhausted all of the candidate splits

2. Accuracy of Decision Tree

Definition of Accuracy and Error

- Given a set D of samples and a trained model M, the accuracy is the percentage of correctly labeled samples. That is,

$$Accuracy(D, M) = \frac{|\{M(x) = l_x \mid x \in D\}|}{|D|}$$

Where l_x is the true label of sample x and $M(x)$ gives the predicted label of x by M

- Error is a dual concept of accuracy.

But, what is D?

$$Error(D, M) = 1 - Accuracy(D, M)$$

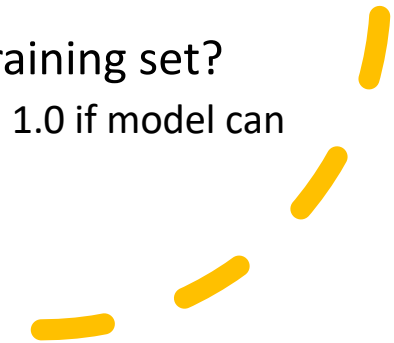
How can we assess the accuracy of a tree?

D = training dataset

- Can we just calculate the fraction of training instances that are correctly classified?

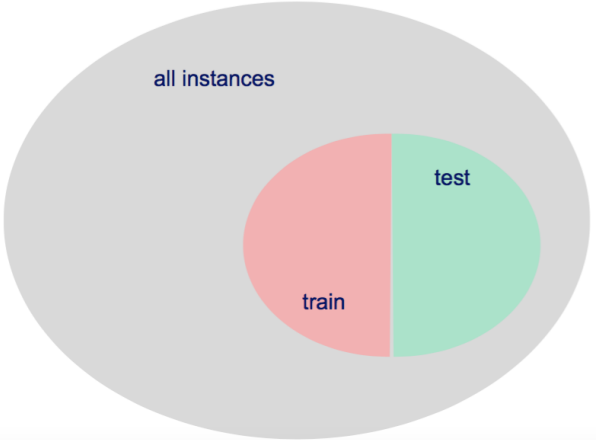
Ground truth

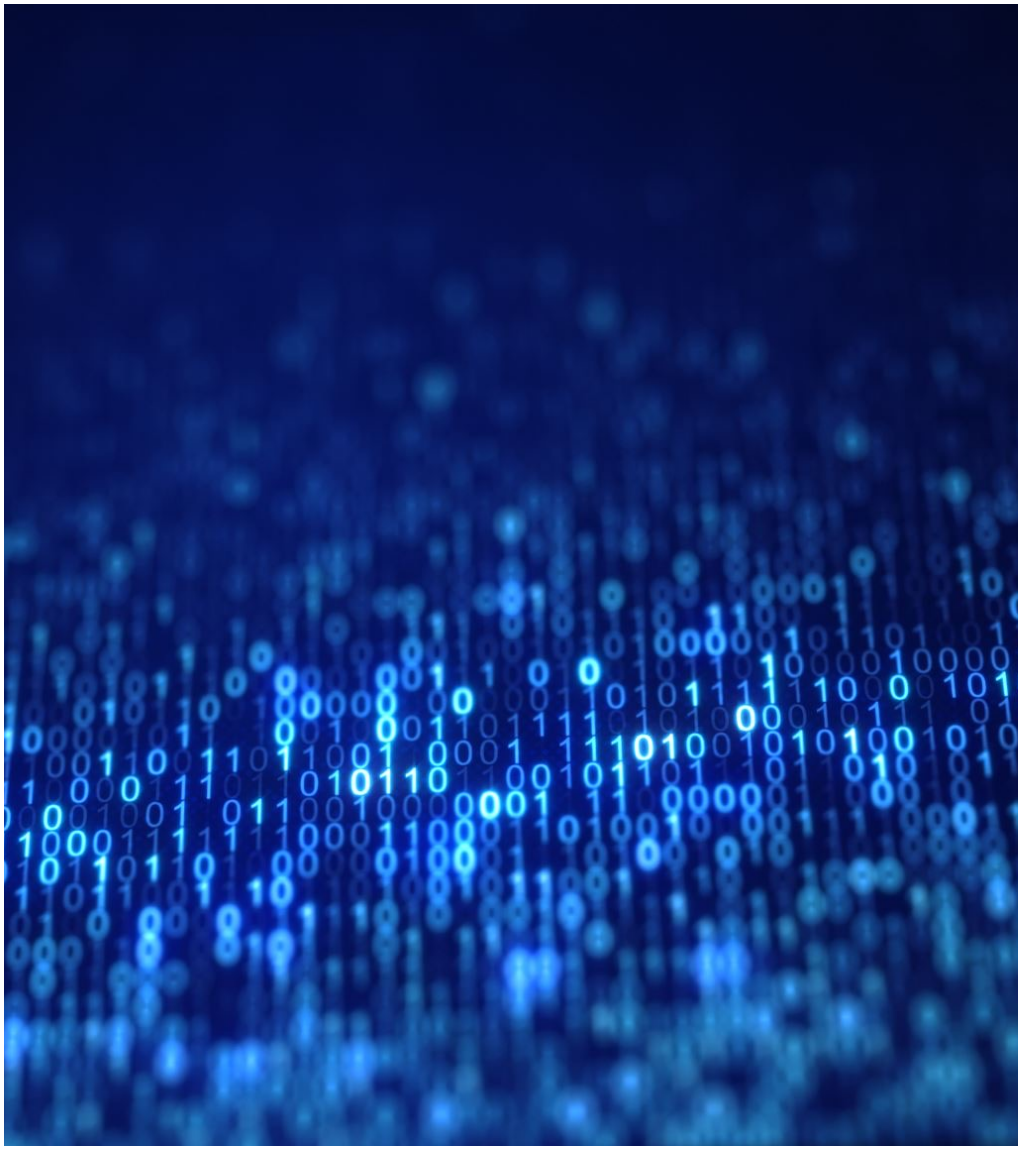
- Consider a problem domain in which instances are assigned labels at random with $P(Y = t) = 0.5$
  - how accurate would a learned decision tree be on previously unseen instances?
    - Can never reach 1.0.
  - how accurate would it be on its training set?
    - Can be arbitrarily close to, or reach, 1.0 if model can be very large.



How can we assess the accuracy of a tree?

- to get an unbiased estimate of a learned model's accuracy, we must use a set of instances that are held-aside during learning
- this is called a *test set*

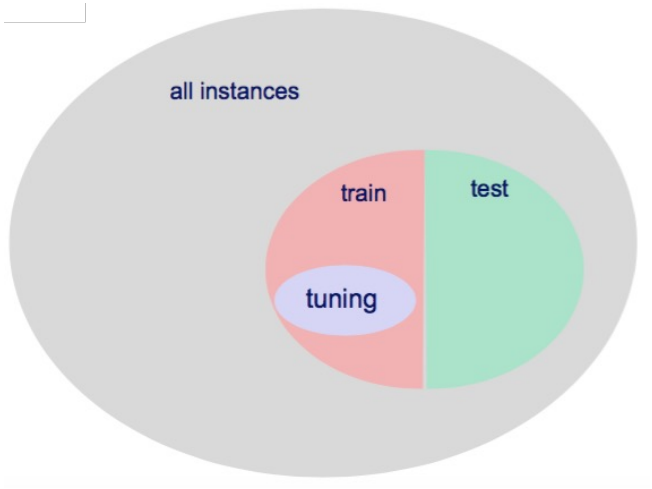




3. Pruning and Validation Dataset

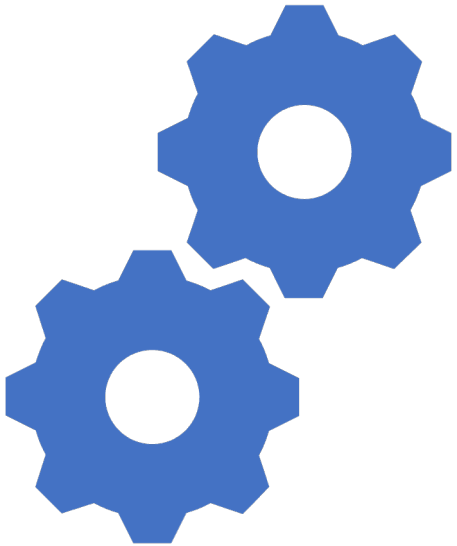
Stopping criteria

- We should form a leaf when
  - all of the given subset of instances are of the same class
  - we've exhausted all of the candidate splits
- Is there a reason to stop earlier, or to prune back the tree?



Pruning in C4.5

- Split given data into training and *validation* (*tuning*) sets
- A *validation set* (a.k.a. *tuning set*) is a subset of the training set that is held aside
  - not used for primary training process (e.g. tree growing)
  - but used to select among models (e.g. trees pruned to varying degrees)

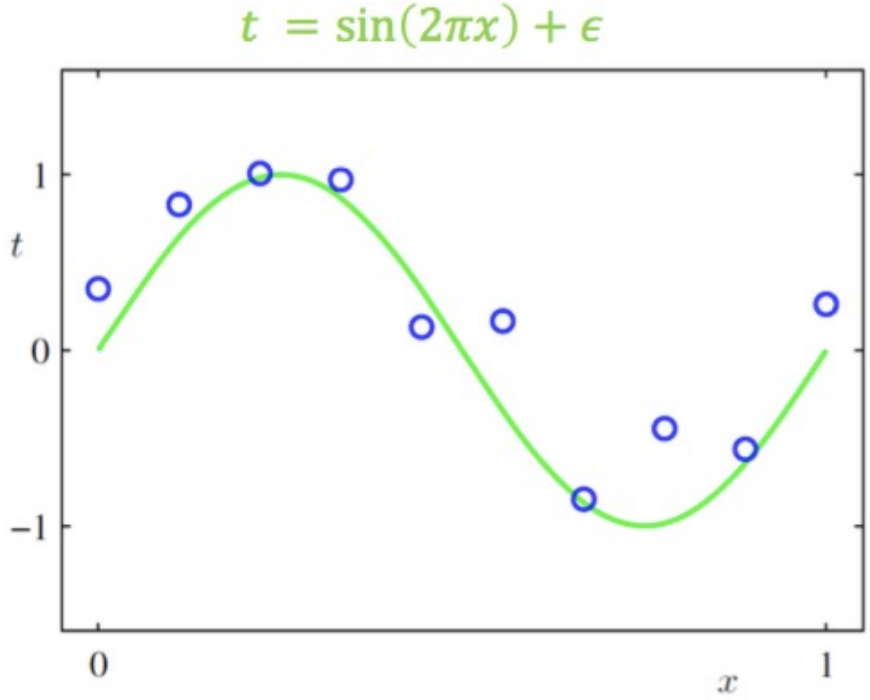


Pruning in C4.5

- Split given data into training and *validation* (*tuning*) sets
- Grow a complete tree
- do until further pruning is harmful
  - evaluate impact on tuning-set accuracy of pruning each node
  - greedily remove the one that least reduces tuning-set accuracy

4. Overfitting

Example 3: regression using polynomial

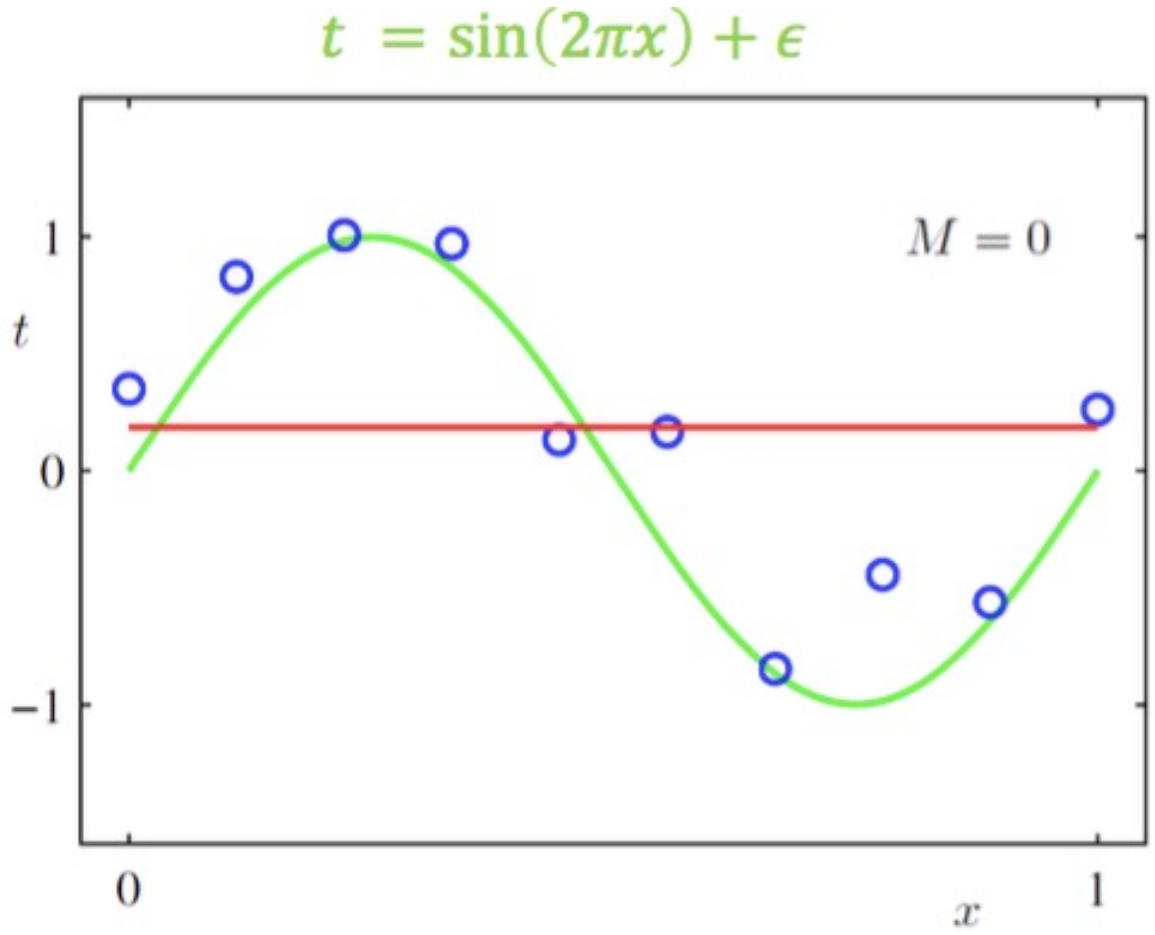


Regression using polynomial of degree M

$y = w_M x^M + w_{M-1} x^{M-1} + \ldots + w_1 x + w_0$

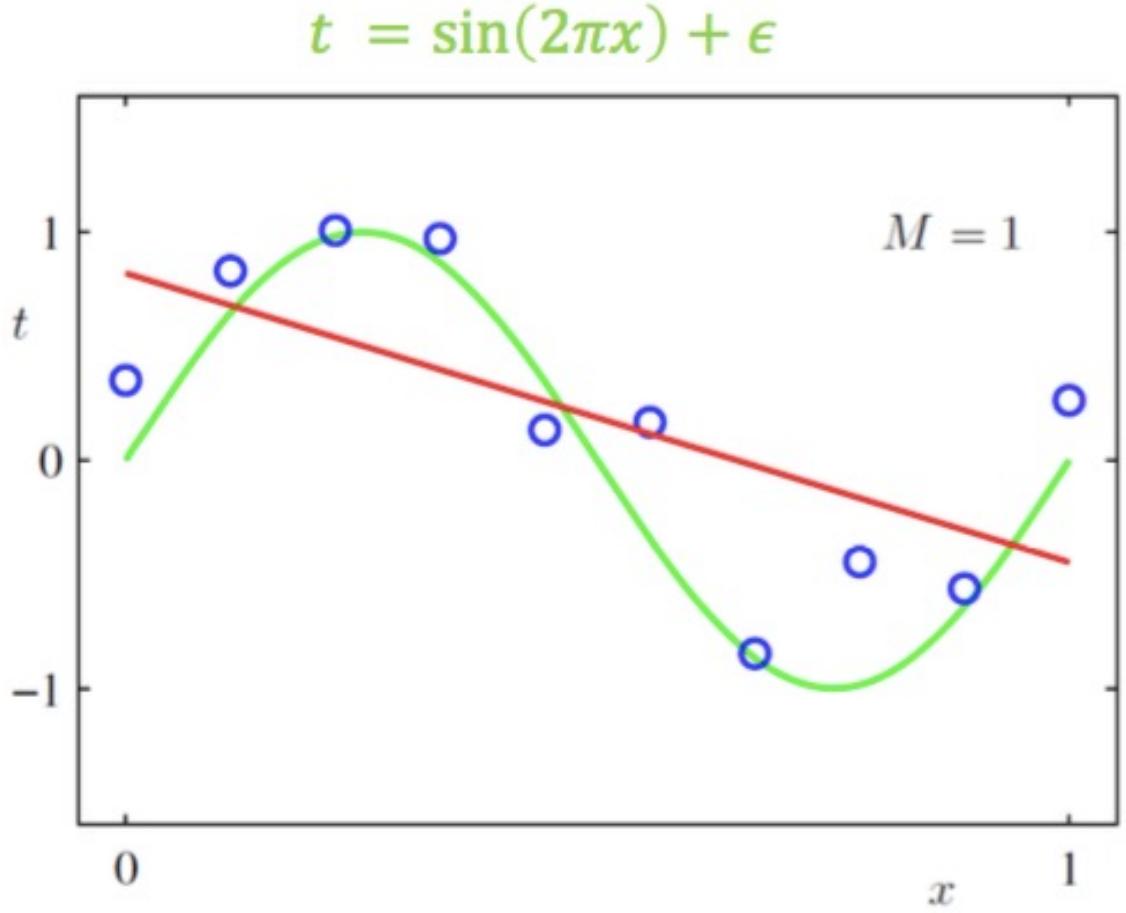
Example 3: regression using polynomial

y = c



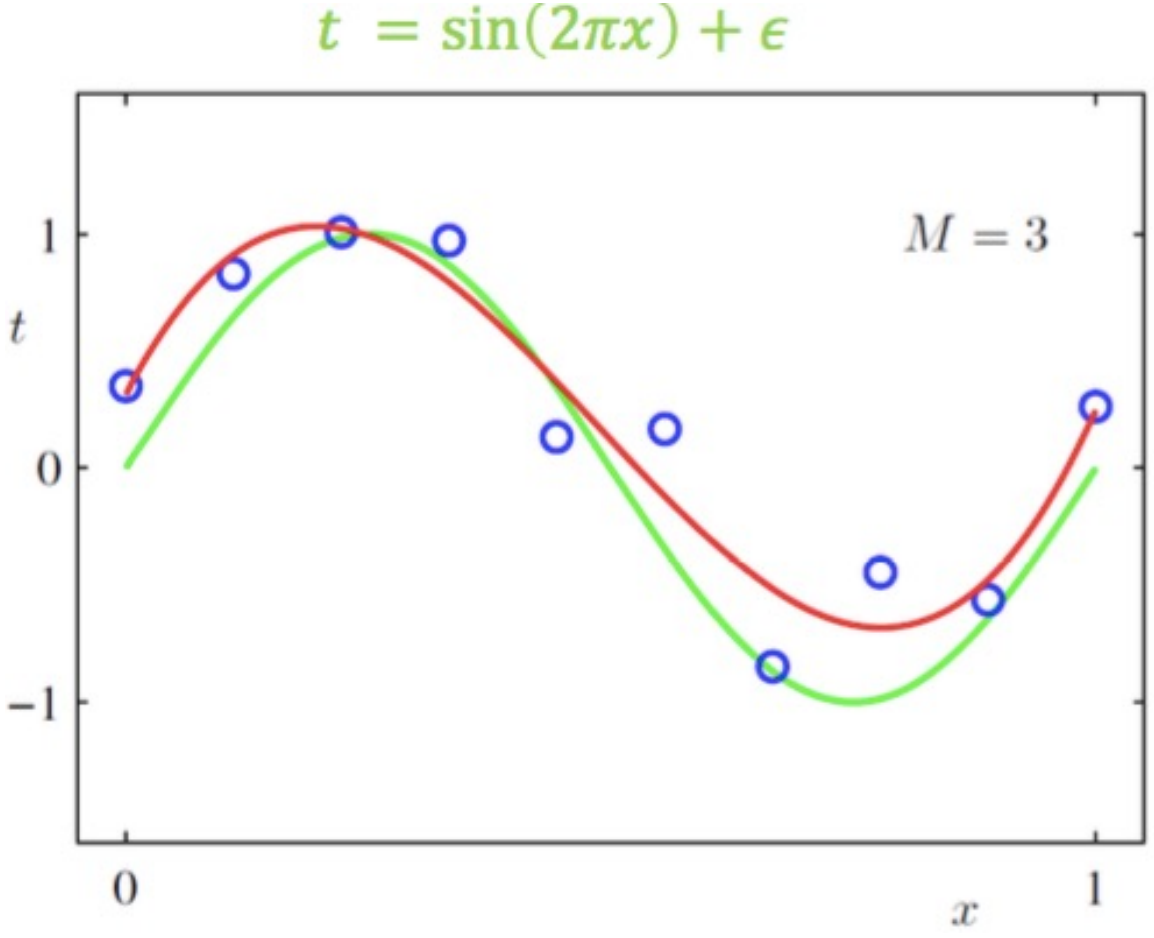
Example 3: regression using polynomial

$y = w_1 x + w_0$



Example 3: regression using polynomial

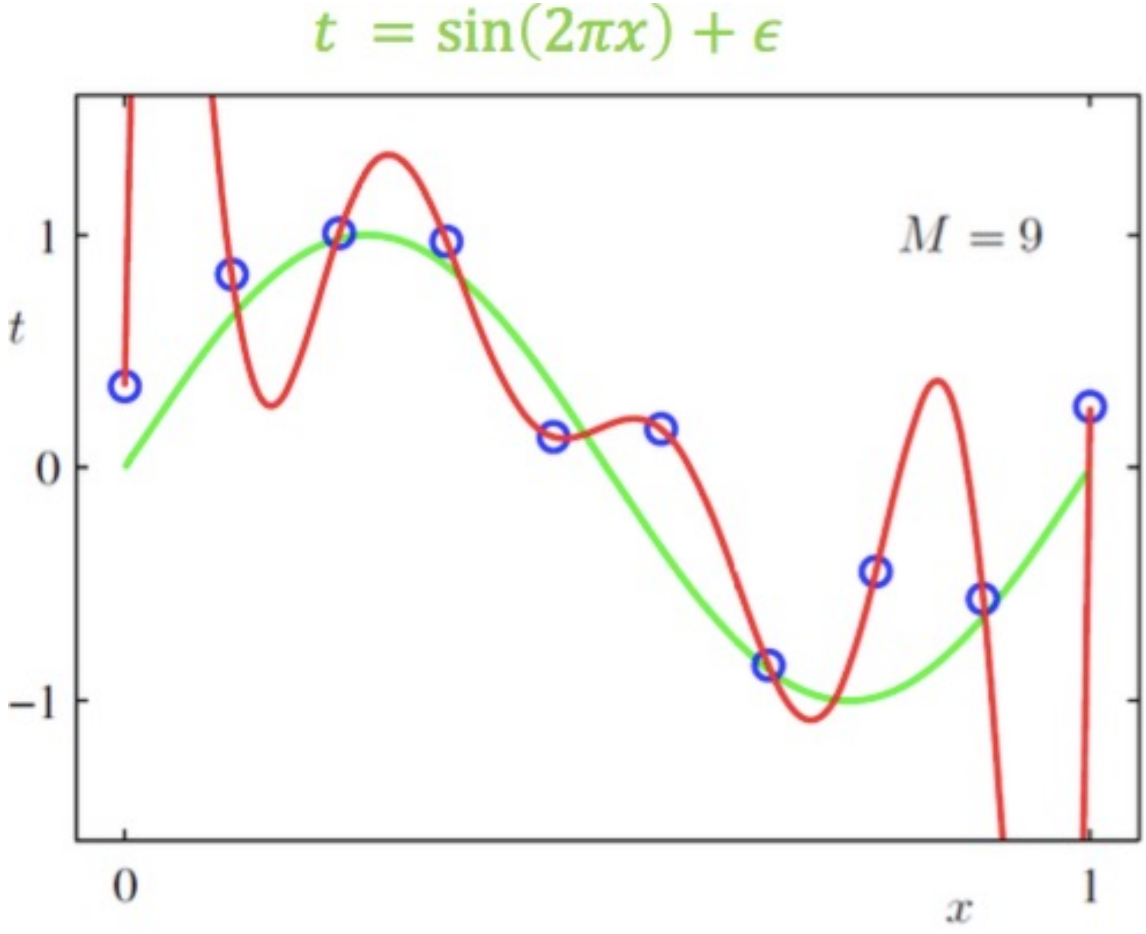
$y = w_3x^3 + w_2x^2 + w_1x + w_0$



Example 3: regression using polynomial

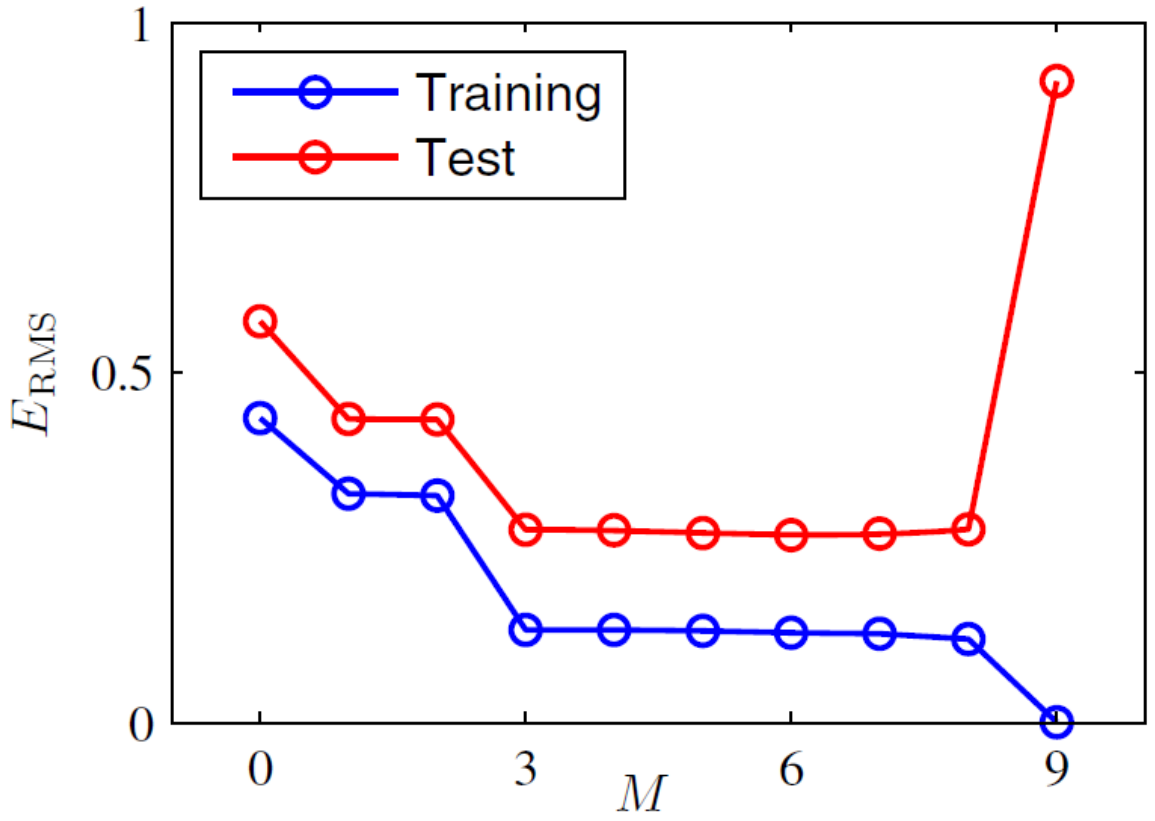
$y = w_9x^9 + w_8x^8 + \ldots + w_1x + w_0$

Overfits, why?

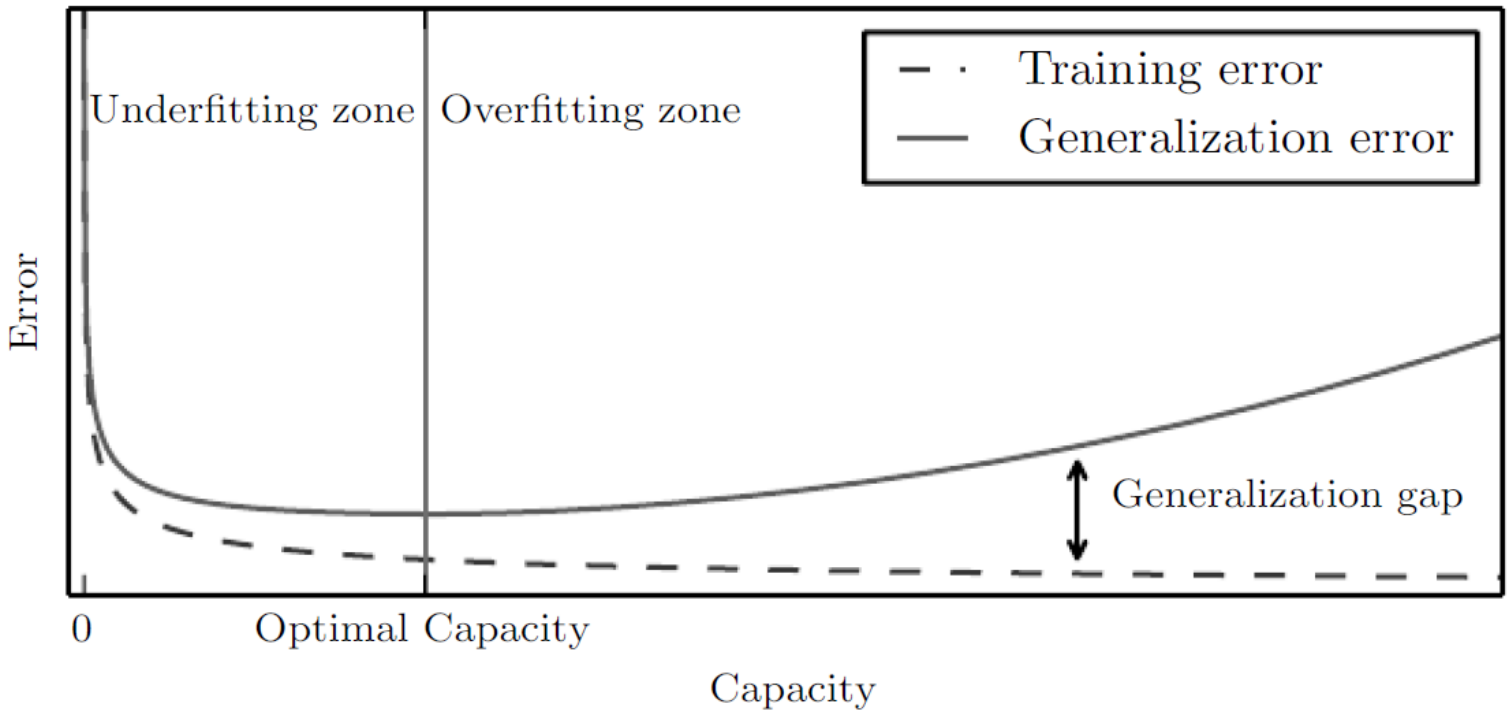


Example: regression using polynomial

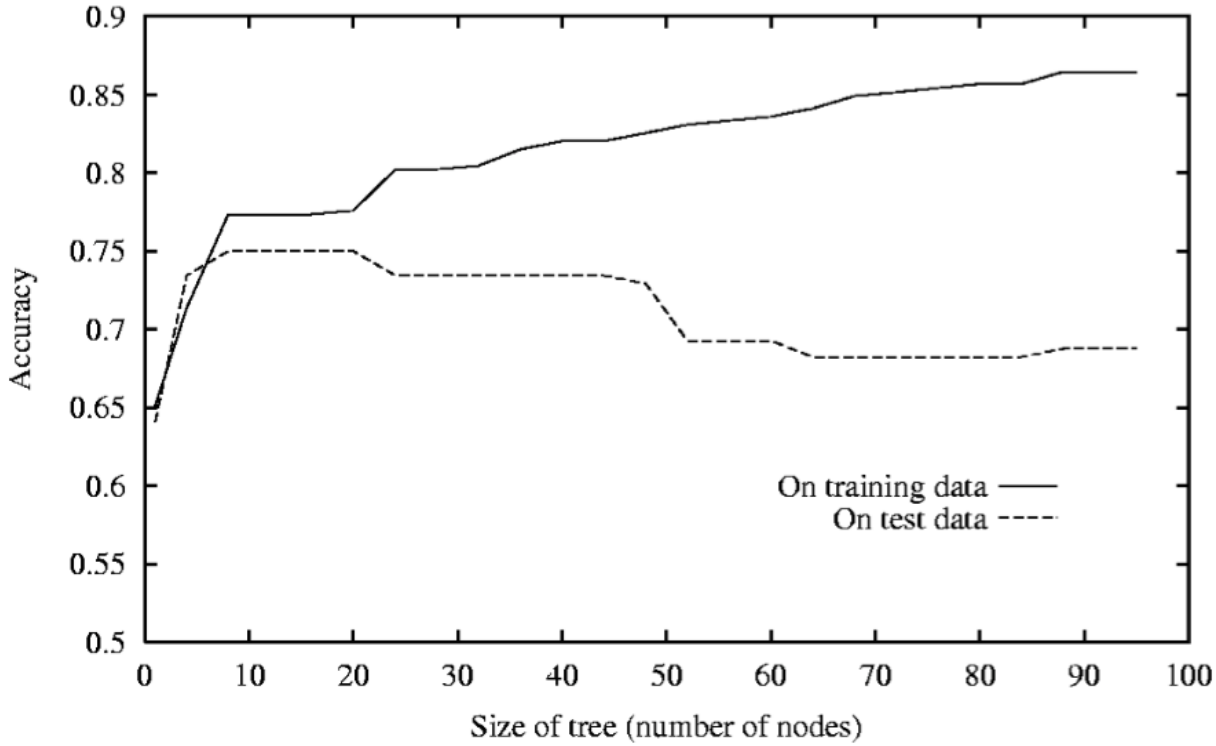
RMS: root mean square, i.e., the square root of the mean square



General phenomenon



Overfitting in decision trees



Prevent overfitting

Will have examples in decision tree later for these two cases

- Cause: training error and expected error are different
  - there may be noise in the training data
  - training data is of limited size, resulting in difference from the true distribution
  - the larger the hypothesis class, the easier to find a hypothesis that fits the difference between the training data and the true distribution

Small training dataset

Large model

- How to reduce overfitting:
  - cleaner training data help!
  - more training data help!
  - throwing away unnecessary hypotheses helps! (Occam's Razor)

5. Overfitting in Decision Tree

Overfitting

- Consider error of model M over
  - training data: $Error(D_{training}, M)$
  - entire distribution of data: $Error(D_{true}, M)$
- Model $M \in H$ *overfits* the training data if there is an alternative model $M' \in H$ such that

$$Error(D_{training}, M) < Error(D_{training}, M')$$

Perform better on training dataset

$$Error(D_{true}, M) > Error(D_{true}, M')$$

Perform worse on true distribution

Example 1: overfitting with noisy data

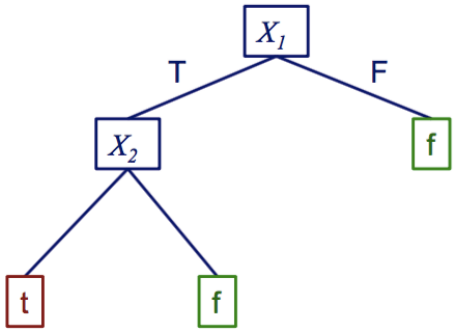
- suppose
  - the target concept is $Y = X_1 \wedge X_2$ Ground truth
  - there is noise in some feature values
  - we're given the following training set

X_1	X_2	X_3	X_4	X_5	...	Y
t	t	t	t	t	...	t
t	t	f	f	t	...	t
t	**f**	t	t	f	...	t
t	f	f	t	f	...	f
t	f	t	f	f	...	f
f	t	t	f	t	...	f

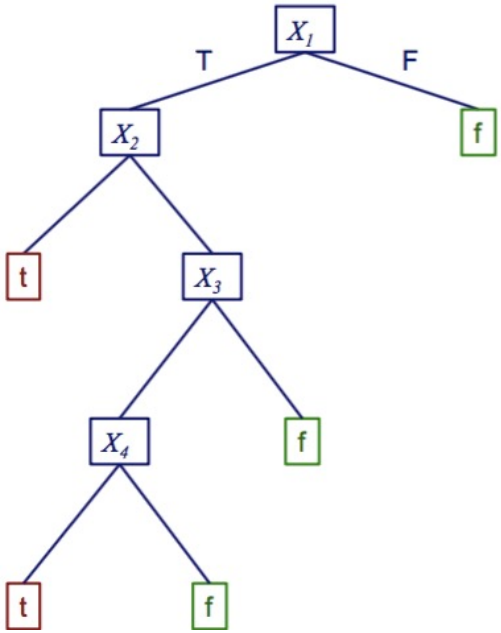
noisy value

Example 1: overfitting with noisy data

correct tree



tree that fits noisy training data



$$Y = X_1 \wedge X_2$$

A noisy data sample:

X_1 = t

X_2 = f

X_3 = t

X_4 = t

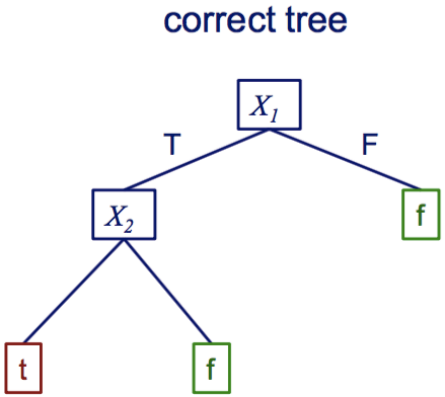
X_5 = f

Y = t

X_1	X_2	X_3	X_4	X_5	...	Y
t	t	t	t	t	...	t
t	t	f	f	t	...	t
t	**f**	t	t	f	...	t
t	f	f	t	f	...	f
t	f	t	f	f	...	f
f	t	t	f	t	...	f

noisy value

Example 1: overfitting with noisy data



What is the accuracy?

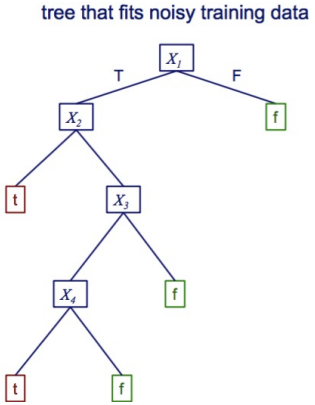
- Accuracy($D_{training}$,M) = 5/6
- Accuracy(D_{true},M) = 100%

$$Y = X_1 \wedge X_2$$

X_1	X_2	X_3	X_4	X_5	...	Y
t	t	t	t	t	...	t
t	t	f	f	t	...	t
t	f	t	t	f	...	t
t	f	f	t	f	...	f
t	f	t	f	f	...	f
f	t	t	f	t	...	f

noisy value

Example 1: overfitting with noisy data



$$Y = X_1 \wedge X_2$$

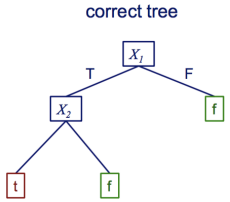
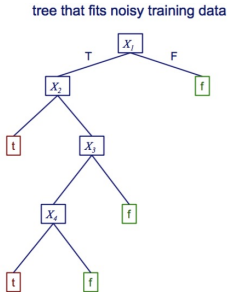
X_1	X_2	X_3	X_4	X_5	...	Y
t	t	t	t	t	...	t
t	t	f	f	t	...	t
t	f	t	t	f	...	t
t	f	f	t	f	...	f
t	f	t	f	f	...	f
f	t	t	f	t	...	f

noisy value

What is the accuracy?

- Accuracy($D_{training}$,M) = 100%
- Accuracy(D_{true},M) < 100%

Example 1: overfitting with noisy data

		Training set accuracy	True accuracy
M_1		5/6	100%
M_2		100%	< 100 %

M_2 is overfitting!

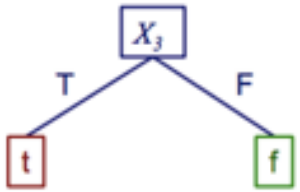
Example 2: overfitting with noise-free data

- suppose
  - the target concept is $Y = X_1 \wedge X_2$
  - $P(X_3 = t) = 0.5$ for both classes
  - $P(Y = t) = 0.66$
  - we're given the following training set

Ground truth

X_1	X_2	X_3	X_4	X_5	...	Y
t	t	t	t	t	...	t
t	t	t	f	t	...	t
t	t	t	t	f	...	t
t	f	f	t	f	...	f
f	t	f	f	t	...	f

Example 2: overfitting with noise-free data

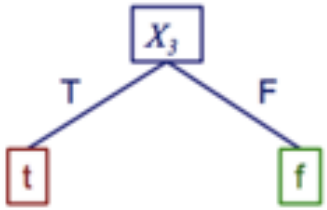


M_1

M_2

X_1	X_2	X_3	X_4	X_5	...	Y
t	t	t	t	t	...	t
t	t	t	f	t	...	t
t	t	t	t	f	...	t
t	f	f	t	f	...	f
f	t	f	f	t	...	f

Example 2: overfitting with noise-free data



$$Y = X_1 \wedge X_2$$

$P(X_3 = t) = 0.5$

$P(Y=t) = 0.66$

What is the accuracy?

- Accuracy($D_{training}$,M) = 100%
- Accuracy(D_{true},M) = 50%

(X3, Y):
(t, t) = 0.5*0.66 (correct)
(t, f) = 0.5*0.33 (fail)
(f, t) = 0.5 *0.66 (fail)
(f, f) = 0.5 *0.33 (correct)

So, we have
50% correctness

X_1	X_2	X_3	X_4	X_5	...	Y
t	t	t	t	t	...	t
t	t	t	f	t	...	t
t	t	t	t	f	...	t
t	f	f	t	f	...	f
f	t	f	f	t	...	f

Example 2: overfitting with noise-free data

t

$$Y = X_1 \wedge X_2$$

$P(X_3 = t) = 0.5$

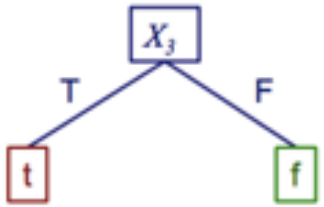
$P(Y=t) = 0.66$

What is the accuracy?

- Accuracy($D_{training}$,M) = 60%
- Accuracy(D_{true},M) = 66%

X_1	X_2	X_3	X_4	X_5	...	Y
t	t	t	t	t	...	t
t	t	t	f	t	...	t
t	t	t	t	f	...	t
t	f	f	t	f	...	f
f	t	f	f	t	...	f

Example 2: overfitting with noise-free data

		Training set accuracy	True accuracy	
M_1		100%	50%	M_1 is overfitting!
M_2		60%	66%	

because the training set is a limited sample, there might be (combinations of) features that are correlated with the target concept by chance

Avoiding overfitting in DT learning

The stopping criterion utilizing validation dataset

- two general strategies to avoid overfitting
  1. *early stopping:* stop if further splitting not justified by a statistical test
     - Quinlan's original approach in ID3
  2. *post-pruning*: grow a large tree, then prune back some nodes
     - more robust to myopia of greedy tree learning