Machine Learning 2007: Lecture 6

Instructor: Tim van Erven (Tim.van.Erven@cwi.nl)

Website: `www.cwi.nl/˜erven/teaching/0708/ml/`

October 11, 2007

Overview

- **Organisational Matters**
- Least Squares Regression with Polynomials
- Train and Test Sets
- Overfitting
  - Overfitting in Prediction, Regression and Classification
  - Complexity of a Hypothesis
- Pruning Decision Trees
  - Reduced Error Pruning
  - Rule Post-Pruning

Course Organisation

Organisation of the Exams:

- October 25: Intermediate Exam
- December 20: Final Exam
  You can choose between:
  - Normal version: Questions about material covered after the intermediate exam. The intermediate exam counts for 20% of your grade. The final exam counts for 40%.
  - Resit version: Questions about all material of the course. The intermediate exam does not count anymore. The final exam counts for 60%.
- Resit: Questions about all material of the course. Your intermediate exam does not count anymore. The final exam counts for 60%.

Other: There will be no lecture on December 5.

The Intermediate Exam

Topics on the Intermediate Exam:

- Everything covered in the first six lectures
- Today is the sixth lecture.
- Study: Chapters 1,2,3 of Mitchell and the corresponding slides

The Intermediate Exam

Topics on the Intermediate Exam:

- Everything covered in the first six lectures
- Today is the sixth lecture.
- Study: Chapters 1,2,3 of Mitchell and the corresponding slides

Organisation of the Intermediate Exam:

- The intermediate exam will most likely be in a different (larger) room than 04A05.
- I will e-mail you an announcement when the room is known, and put a notice on the website.
- This announcement will contain instructions about enrolling.
- (For example, whether enrolling is necessary; Contrary to earlier statements, it might be.)

This Lecture versus Mitchell

Mitchell:

- Read: Chapter 3 of Mitchell.

This Lecture:

- Least squares regression with polynomials is not in Mitchell.
- Slightly different definition of overfitting than in Mitchell.
- More examples of overfitting than Mitchell (also in prediction and regression).
- More discussion of the complexity of a hypothesis than Mitchell.

Overview

- Organisational Matters
- **Least Squares Regression with Polynomials**
- Train and Test Sets
- Overfitting
    - Overfitting in Prediction, Regression and Classification
    - Complexity of a Hypothesis
- Pruning Decision Trees
    - Reduced Error Pruning
    - Rule Post-Pruning

Least Squares Regression with Polynomials

- We have seen least squares regression with linear functions.
- It selects the **linear function** $h_{\mathbf{w}}$ that minimizes the sum of squared errors (SSE) on the data:

$$\min_{\mathbf{w}} \text{SSE}(D) = \min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2.$$

- Instead of a linear function we will now select a **polynomial function** that minimizes the SSE.
- This is a generalisation: Linear functions are specific kinds of polynomials.

Least Squares Regression with Polynomials

- We have seen least squares regression with linear functions.
- It selects the **linear function** $h_{\mathbf{w}}$ that minimizes the sum of squared errors (SSE) on the data:

$$\min_{\mathbf{w}} \mathsf{SSE}(D) = \min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2.$$

- Instead of a linear function we will now select a **polynomial function** that minimizes the SSE.
- This is a generalisation: Linear functions are specific kinds of polynomials.

Simplifying Assumption:

- We will only consider 1-**dimensional** feature vectors here.
- Generalising to higher dimensional feature vectors is possible, but doesn't give you more insight.

Polynomials

Definition:

A **polynomial** of degree d is a function f of the form

$$f(x) = w_d x^d + w_{d-1} x^{d-1} + \ldots + w_1 x^1 + w_0,$$

where $w_0, \ldots, w_d$ are called the **coefficients** of the polynomial.

Polynomials

Definition:

A **polynomial** of degree d is a function f of the form

$$f(x) = w_d x^d + w_{d-1} x^{d-1} + \ldots + w_1 x^1 + w_0,$$

where $w_0, \ldots, w_d$ are called the **coefficients** of the polynomial.

Examples:

$$f(x) = x^2 + 3x + 6 \qquad f(x) = -10$$

$$f(x) = -6x^4 + x^3 - 2x^2 + 3.4x - 1 \qquad f(x) = x^{10} - 7x^7$$

Polynomials

Definition:

A **polynomial** of degree d is a function f of the form

$$f(x) = w_d x^d + w_{d-1} x^{d-1} + \ldots + w_1 x^1 + w_0,$$

where $w_0, \ldots, w_d$ are called the **coefficients** of the polynomial.

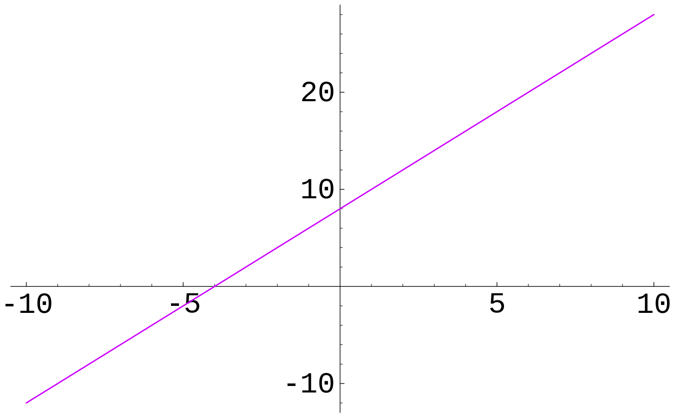
Examples:

$$f(x) = x^2 + 3x + 6 \qquad f(x) = -10$$

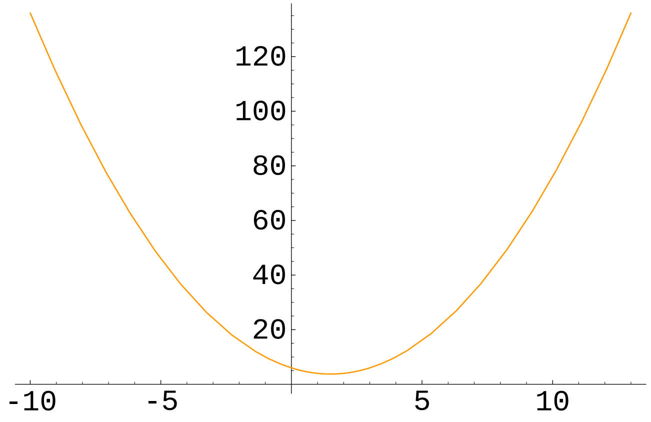
$$f(x) = -6x^4 + x^3 - 2x^2 + 3.4x - 1 \qquad f(x) = x^{10} - 7x^7$$

- Notice that a polynomial is a function of one variable x.
- Linear functions are polynomials of degree 1!
- The set of polynomials of degree d includes all lower order polynomials. (By setting $w_d = 0$ we get a polynomial of degree $d - 1$.)

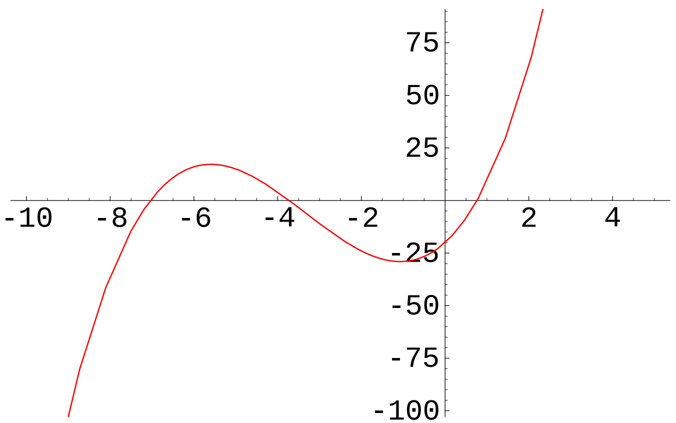
Higher Degrees Are More Complex



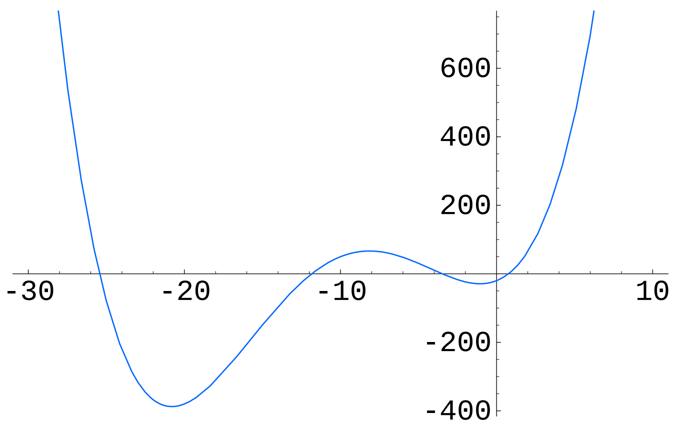
$2x + 8$
1st degree polynomial



$x^2 - 3x + 6$
2nd degree polynomial



$x^3 + 10x^2 + 18x - 20$
3rd degree polynomial



$\frac{1}{40}x^4 + x^3 + 10x^2 + 18x - 20$
4th degree polynomial

Overview

- Organisational Matters
- Least Squares Regression with Polynomials
- **Train and Test Sets**
- Overfitting
    - Overfitting in Prediction, Regression and Classification
    - Complexity of a Hypothesis
- Pruning Decision Trees
    - Reduced Error Pruning
    - Rule Post-Pruning

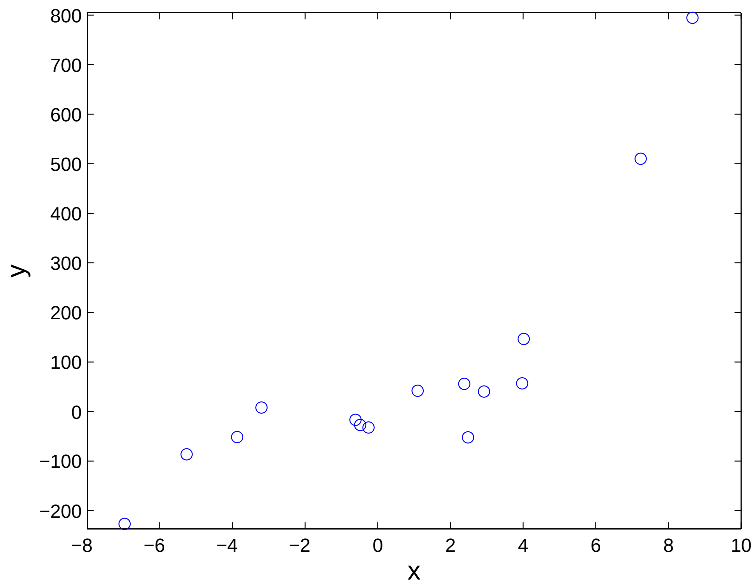
Motivation: Don't Allow Cheating

- How do we evaluate the quality of a hypothesis h that we have learned from data D? Evaluate Error(h, D)?

Train and Test Sets

Motivation: Don't Allow Cheating

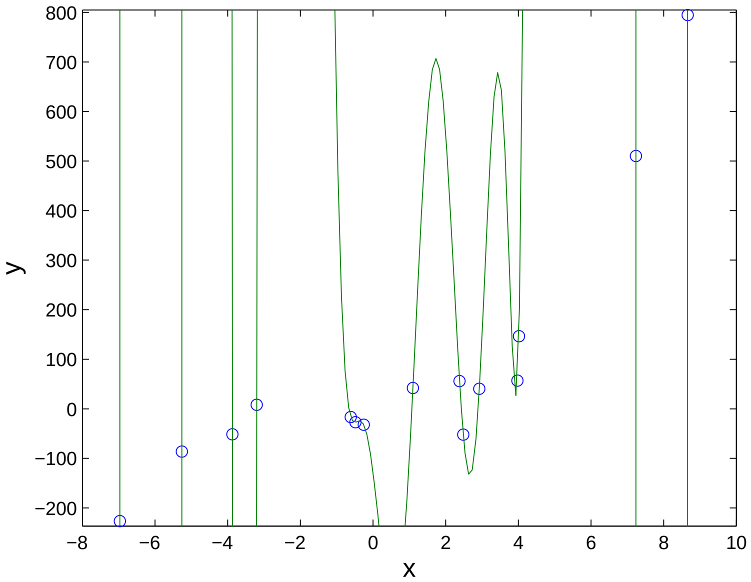
- How do we evaluate the quality of a hypothesis h that we have learned from data D? Evaluate Error(h, D)?
- Maybe h just stores the correct answers for D, but doesn't know how to generalise at all.
- Training and testing on the same data allows cheating!



Train and Test Sets

Motivation: Don't Allow Cheating

- How do we evaluate the quality of a hypothesis h that we have learned from data D? Evaluate Error(h, D)?
- Maybe h just stores the correct answers for D, but doesn't know how to generalise at all.
- Training and testing on the same data allows cheating!



Train and Test Sets

Motivation: Don't Allow Cheating

- How do we evaluate the quality of a hypothesis h that we have learned from data D? Evaluate Error(h, D)?
- Maybe h just stores the correct answers for D, but doesn't know how to generalise at all.
- Training and testing on the same data allows cheating!

Train and Test Sets

- Split the data D into a train set D_{train} and a test set D_{test}.
- **Train set:** data used to train a machine learning algorithm (e.g. to select a hypothesis h)
- **Test set:** data used to evaluate the performance of the algorithm (e.g. by evaluating Error(h,D_{test}))

Validation Set

- **Train set:** data D_{train} used to train a machine learning algorithm (e.g. to select a hypothesis h)
- **Test set:** data D_{test} used to evaluate the performance of the algorithm (e.g. by evaluating Error(h,D_{test}))

Remarks:

- Suppose our machine learning method has some parameters.
- Then we may be tempted to run it with different paremeter values on the train set, and pick the best parameter settings according to the test set.

Validation Set

- **Train set:** data D_{train} used to train a machine learning algorithm (e.g. to select a hypothesis h)
- **Test set:** data D_{test} used to evaluate the performance of the algorithm (e.g. by evaluating Error(h,D_{test}))

Remarks:

- Suppose our machine learning method has some parameters.
- Then we may be tempted to run it with different paremeter values on the train set, and pick the best parameter settings according to the test set.
- This is cheating again: We are putting information about the test set into our algorithm through the parameters.
- Instead, we should split the data into three parts: a train set, a test set, and a **validation set** that will be used to optimize the parameters.

Overview

- Organisational Matters
- Least Squares Regression with Polynomials
- Train and Test Sets
- **Overfitting**
  - Overfitting in Prediction, Regression and Classification
  - Complexity of a Hypothesis
- Pruning Decision Trees
  - Reduced Error Pruning
  - Rule Post-Pruning

Definition

A hypothesis $h \in \mathcal{H}$ **overfits** the train set if there exists a hypothesis $h' \in \mathcal{H}$ such that: h performs better than h' on the **train set**:

$$\text{Error}(h, D_{\text{train}}) < \text{Error}(h', D_{\text{train}}), \tag{1}$$

but h generalises less well than h': For a suffiently large **test set**,

$$\text{Error}(h, D_{\text{test}}) > \text{Error}(h', D_{\text{test}}). \tag{2}$$

Definition

A hypothesis $h \in \mathcal{H}$ **overfits** the train set if there exists a hypothesis $h' \in \mathcal{H}$ such that: h performs better than h' on the **train set**:

$$\text{Error}(h, D_{\text{train}}) < \text{Error}(h', D_{\text{train}}), \tag{1}$$

but h generalises less well than h': For a suffiently large **test set**,

$$\text{Error}(h, D_{\text{test}}) > \text{Error}(h', D_{\text{test}}). \tag{2}$$

Remarks:

- Interpretation: On the data that we have (D_{train}) it looks as though h is very good, but in reality h generalises poorly.
- One of the **main problems** in machine learning: How to avoid overfitting.
- The second equation in the definition is slightly different from Mitchell, but the idea is the same.

Overview

- Organisational Matters
- Least Squares Regression with Polynomials
- Train and Test Sets
- Overfitting
  - **Overfitting in Prediction, Regression and Classification**
  - Complexity of a Hypothesis
- Pruning Decision Trees
  - Reduced Error Pruning
  - Rule Post-Pruning

Overfitting in Prediction

The Dice Prediction Game (with psychic students):

- $D_{\text{train}} =$ two throws of a die.
- You have to predict the outcomes. (Each student is a hypothesis.)
- Write your predictions down like this:

	First Throw	Second Throw
Prediction	...	...

Overfitting in Prediction

The Dice Prediction Game (with psychic students):

- $D_{\text{train}} =$ two throws of a die.
- You have to predict the outcomes. (Each student is a hypothesis.)
- Write your predictions down like this:

	First Throw	Second Throw
Prediction	...	...

Should We Trust Psychics?

- $D_{\text{test}} =$ hundred more throws of the same die.
- Are the students who predicted the first two throws correctly more likely to predict the last hundred throws correctly?

Overfitting in Prediction

The Dice Prediction Game (with psychic students):

- $D_{\text{train}} =$ two throws of a die.
- You have to predict the outcomes. (Each student is a hypothesis.)
- Write your predictions down like this:

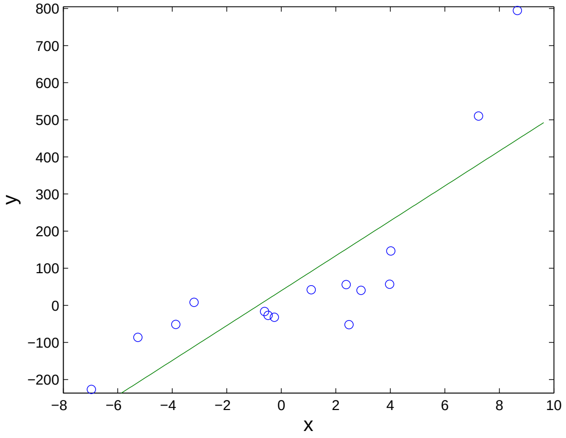
	First Throw	Second Throw
Prediction	...	...

Should We Trust Psychics?

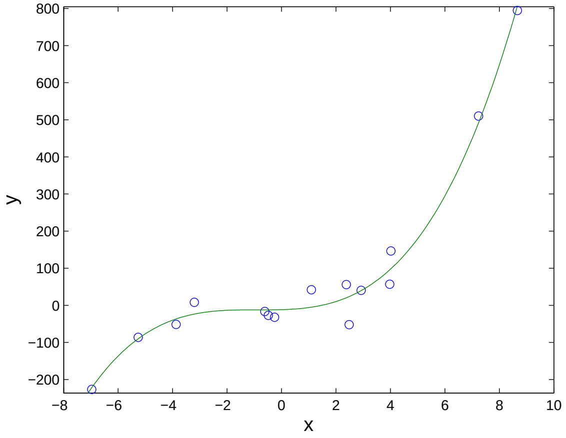
- $D_{\text{test}} =$ hundred more throws of the same die.
- Are the students who predicted the first two throws correctly more likely to predict the last hundred throws correctly?
- Clearly not.
- The **reason for overfitting**: When selecting hypotheses (students) from a **large hypothesis space** (class), some will fit the train set well by coincidence.

Overfitting in Regression

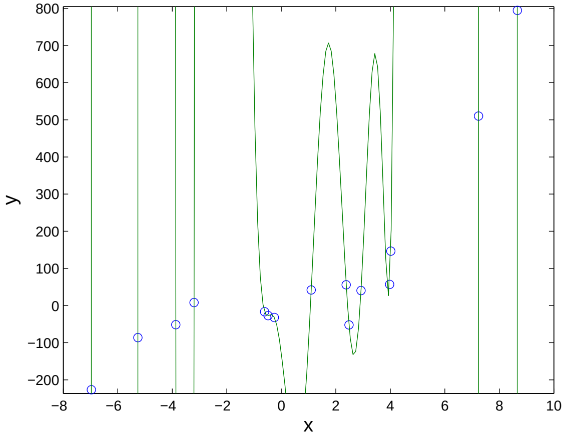
Minimizing the SSE for different degree polynomials:



Linear Function (Simple)



Third Degree Polynomial (Intermediate)



14th Degree polynomial (Complex)

- We consider **three different hypothesis spaces**.
- Although the 14th degree polynomial achieves SSE$(D) = 0$, it will generalise poorly: It overfits the data.

The ID3 *Algorithm Reminder*

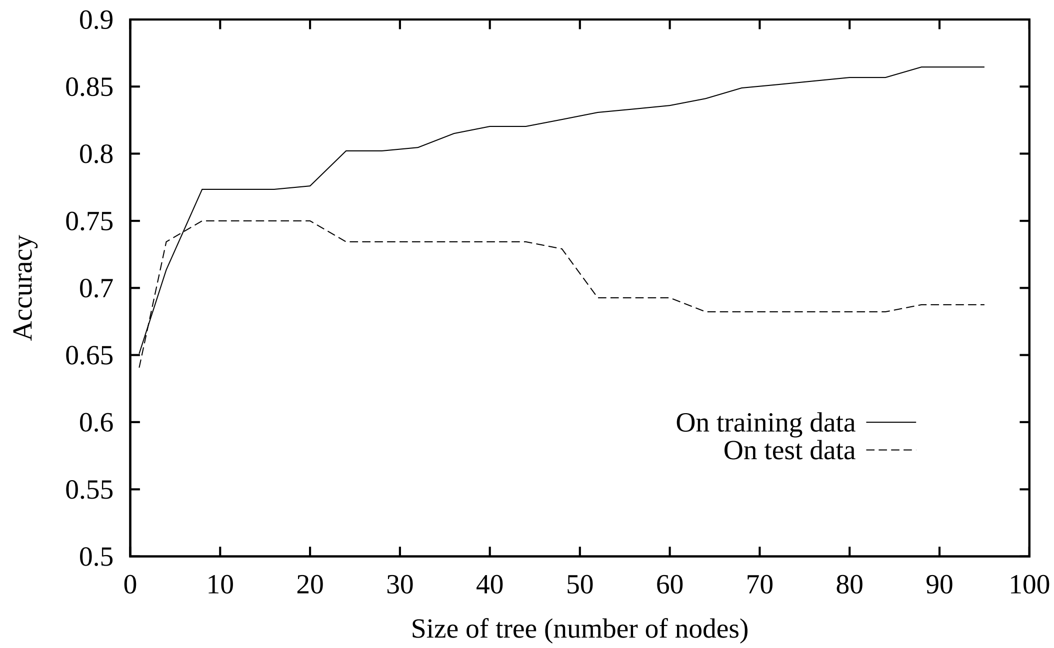
General:

- Learns a decision tree from data.
- Hence does classification.

Main Ideas:

1. Start by selecting a root attribute for the tree.
2. Then grow the tree by adding more and more attributes to it.
3. Stop growing the tree when it is consistent with all the data.

Overfitting with ID3 in Classification



- **This graph is characteristic of overfitting!**
- If the complexity of the selected hypothesis is too low, then increasing complexity increases performance on the test set.
- But if we allow too complex hypotheses, then performance on the train set keeps going up, but generalisation performance will go down.

Overview

- Organisational Matters
- Least Squares Regression with Polynomials
- Train and Test Sets
- Overfitting
  - Overfitting in Prediction, Regression and Classification
  - **Complexity of a Hypothesis**
- Pruning Decision Trees
  - Reduced Error Pruning
  - Rule Post-Pruning

The Complexity of a Hypothesis

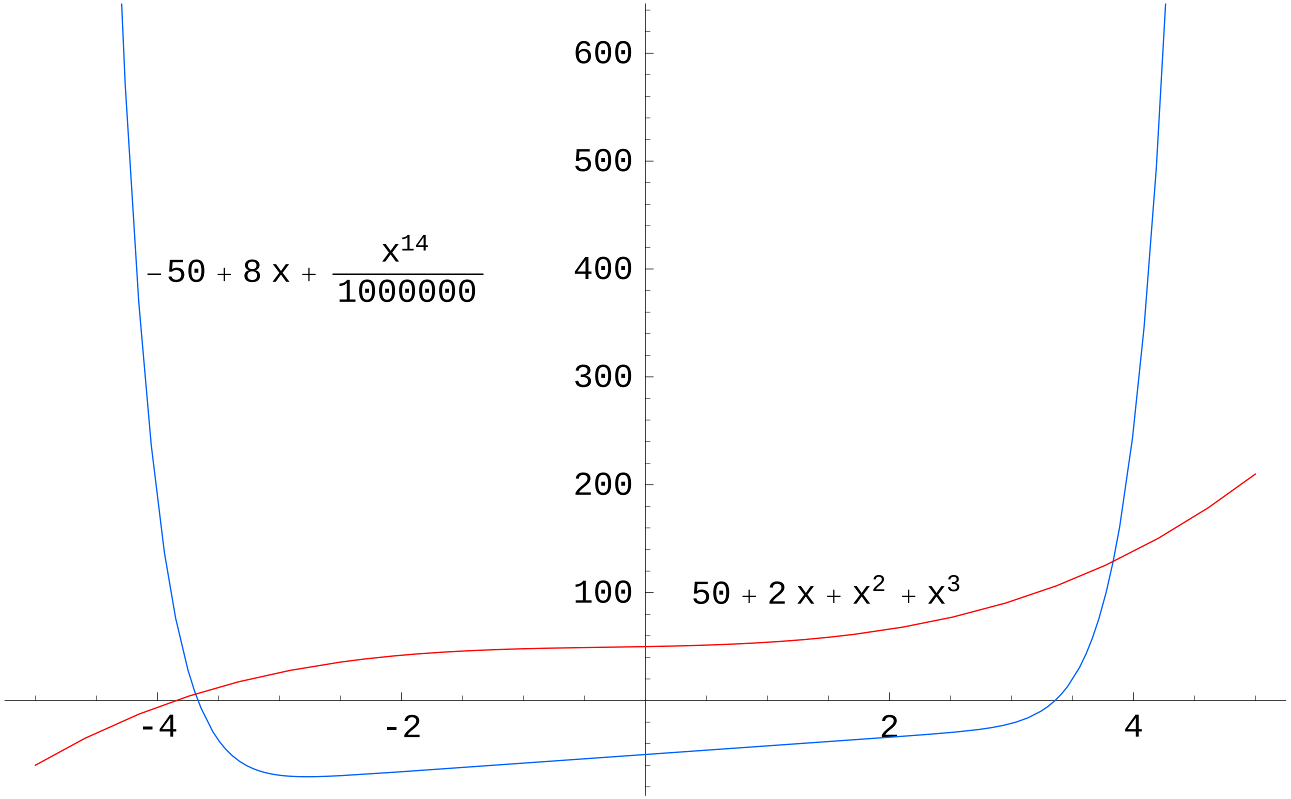
What is Complex?

- "A 14th degree polynomial is more complex than a 3rd degree polynomial." But why should it be?

The Complexity of a Hypothesis

What is Complex?

- "A 14th degree polynomial is more complex than a 3rd degree polynomial." But why should it be?
- For example, compare $10^{-6}x^{14} + 8x - 50$ to $x^3 + x^2 + 2x + 50$. Why should the former be more complex than the latter?



The Complexity of a Hypothesis

What is Complex?

- "A 14th degree polynomial is more complex than a 3rd degree polynomial." But why should it be?
- For example, compare $10^{-6}x^{14} + 8x - 50$ to $x^3 + x^2 + 2x + 50$. Why should the former be more complex than the latter?

The Complexity of a Hypothesis:

- **A hypothesis is complex if it only appears as a member of a large hypothesis space.**
- Hence 14th degree polynomials are more complex than 3rd degree polynomials, because they are members only of a larger hypothesis space.

The Complexity of a Hypothesis

What is Complex?

- "A 14th degree polynomial is more complex than a 3rd degree polynomial." But why should it be?
- For example, compare $10^{-6}x^{14} + 8x - 50$ to $x^3 + x^2 + 2x + 50$. Why should the former be more complex than the latter?

The Complexity of a Hypothesis:

- **A hypothesis is complex if it only appears as a member of a large hypothesis space.**
- Hence 14th degree polynomials are more complex than 3rd degree polynomials, because they are members only of a larger hypothesis space.
- So **complexity depends on which hypothesis spaces we are considering**.
- If we also consider $\mathcal{H} = \{10^{-6}x^{14} - 8x + w_0 \mid w_0 \in \mathbb{R}\}$, then **some** 14th degree hypotheses are not very complex, because they appear in this relatively small $\mathcal{H}$.

Overview

- Organisational Matters
- Least Squares Regression with Polynomials
- Train and Test Sets
- Overfitting
  - Overfitting in Prediction, Regression and Classification
  - Complexity of a Hypothesis
- **Pruning Decision Trees**
  - Reduced Error Pruning
  - Rule Post-Pruning

Smaller Trees in ID3

- We have seen that ID3 may grow a tree that is too complex/large.
- There are two ways to avoid this:

1. Stop growing the tree earlier, before it perfectly classifies all examples in the train set.
2. First grow the full tree, then **post-prune** it.

Smaller Trees in ID3

- We have seen that ID3 may grow a tree that is too complex/large.
- There are two ways to avoid this:
  1. Stop growing the tree earlier, before it perfectly classifies all examples in the train set.
  2. First grow the full tree, then **post-prune** it.
- Although the first approach seems more direct, it is very difficult.
  - I suspect this is because in each recursion of the ID3 algorithm, the data is split up.
  - The decision to stop growing is made after a number of these splits.
  - Thus it is based on a tiny fraction of the data.
- The second approach (pruning) has been found to be more successful in practice.

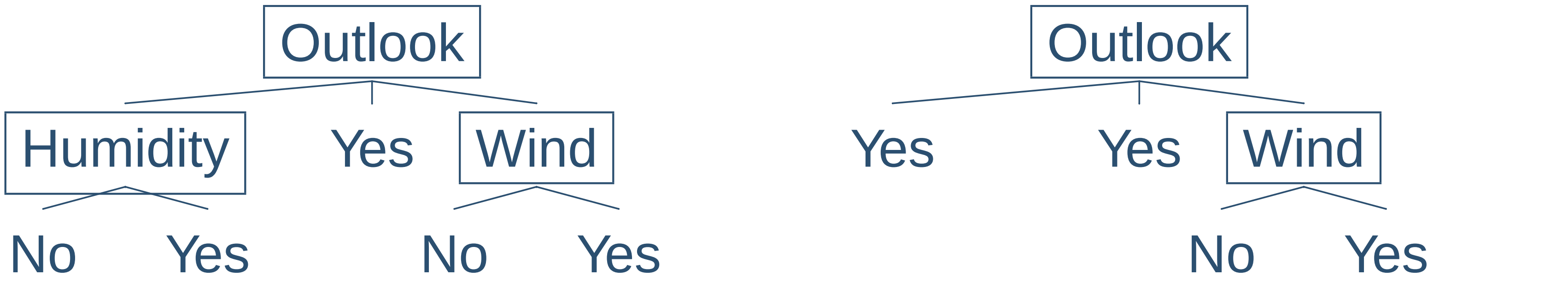
Overview

- Organisational Matters
- Least Squares Regression with Polynomials
- Train and Test Sets
- Overfitting
  - Overfitting in Prediction, Regression and Classification
  - Complexity of a Hypothesis
- Pruning Decision Trees
  - **Reduced Error Pruning**
  - Rule Post-Pruning

Reduced Error Pruning

- Use a validation set to decide which nodes to remove (prune):

Removing a Node:



Reduced Error Pruning:

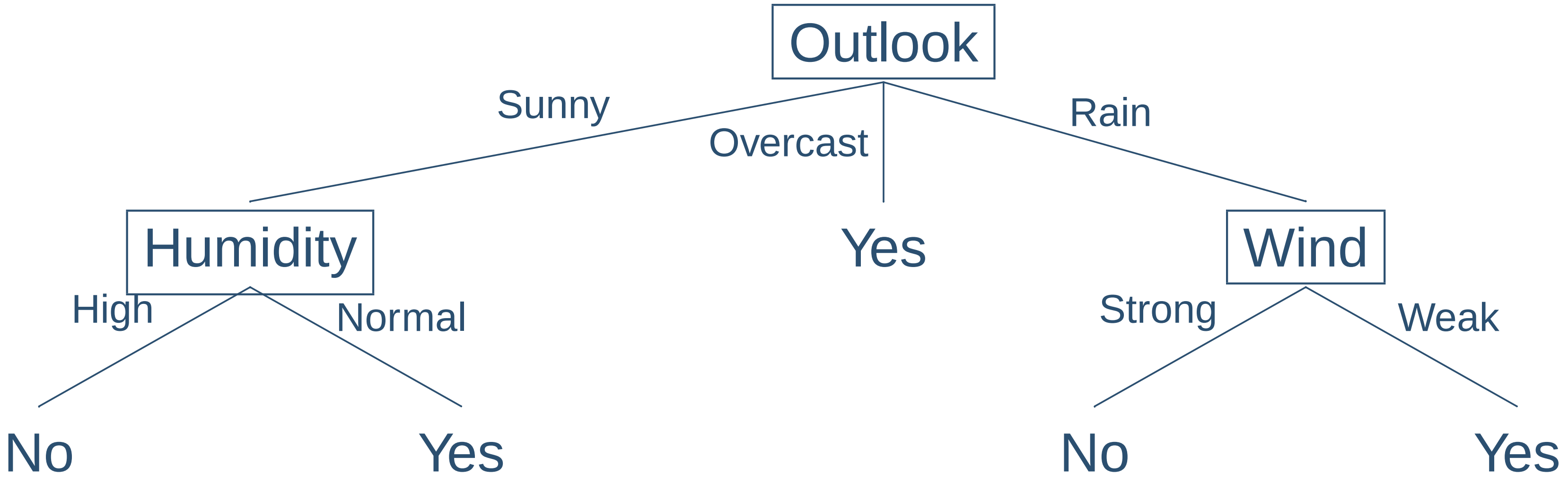
while it increases accuracy on the validation set. **do**
  Remove node that most improves accuracy on validation set.
end while

Overview

- Organisational Matters
- Least Squares Regression with Polynomials
- Train and Test Sets
- Overfitting
  - Overfitting in Prediction, Regression and Classification
  - Complexity of a Hypothesis
- Pruning Decision Trees
  - Reduced Error Pruning
  - **Rule Post-Pruning**

Turning a Tree into a Set of Decision Rules

Tree:



Equivalent Set of Rules:

If Outlook=Sunny $\wedge$ Humidity=High **Then** PlayTennis=No
If Outlook=Sunny $\wedge$ Humidity=Normal **Then** PlayTennis=Yes
If Outlook=Overcast **Then** PlayTennis=Yes
If Outlook=Rain $\wedge$ Wind=Strong **Then** PlayTennis=No
If Outlook=Rain $\wedge$ Wind=Weak **Then** PlayTennis=Yes

- The **preconditions** of a rule are the conditions before ‘then’.

Rule Post-Pruning

Rule Post-Pruning

1: Run ID3 to grow the decision tree from the train set.
2: Convert the tree into an equivalent set of decision rules.
3: Prune (generalise) each rule: Remove any preconditions such that the resulting rule has improved estimated accuracy.
4: Sort rules by their estimated accuracy.

Example of Removing a Precondition:

- Before:
  If Outlook=Sunny $\wedge$ Humidity=High **Then** PlayTennis=No
- After:
  If Humidity=High **Then** PlayTennis=No

Overview

- Organisational Matters
- Least Squares Regression with Polynomials
- Train and Test Sets
- Overfitting
  - Overfitting in Prediction, Regression and Classification
  - Complexity of a Hypothesis
- Pruning Decision Trees
  - Reduced Error Pruning
  - Rule Post-Pruning

References

- Weisstein, E.W. ”Polynomial.” From MathWorld—A Wolfram Web Resource. http://mathworld.wolfram.com/Polynomial.html
- P.D. Grünwald, “The Minimum Description Length Principle”, MIT Press, 2007
- T.M. Mitchell, “Machine Learning”, McGraw-Hill, 1997