CSC321 Lecture 2
A Simple Learning Algorithm : Linear Regression

Roger Grosse and Nitish Srivastava

January 7, 2015

Outline

In this lecture we will

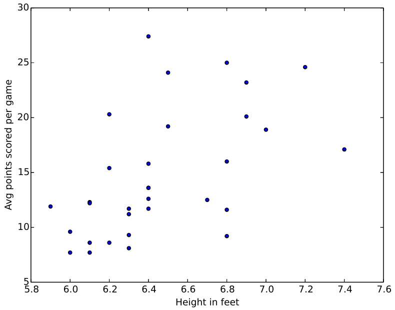
- See a simple example of a machine learning model, linear regression.
- Learn how to formulate a supervised learning problem.
- Learn how to train the model.

It's not a neural net algorithm, but it will provide a lot of useful intuition for algorithms we will cover in this course.

A Machine Learning Problem

Suppose we are given some data about basketball players -

Height in feet	Avg Points Scored Per Game
6.8	9.2
6.3	11.7
6.4	15.8
6.2	8.6
⋮	⋮



What is the predicted number of points scored by a new player who is 6.5 feet tall ?

Formulate as a Supervised Learning Problem

We are given labelled examples (the training set):
Inputs: $\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ - Height in feet.
Targets: $\{t^{(1)}, t^{(2)}, \dots, t^{(N)}\}$ - Avg points scored per game.

Formulate as a Supervised Learning Problem

We are given labelled examples (the training set):
Inputs: $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ - Height in feet.
Targets: $\{t^{(1)}, t^{(2)}, \ldots, t^{(N)}\}$ - Avg points scored per game.

- Choose a model $\equiv$ Make an assumption about the data's behaviour. Let's say we choose -

$$y = wx + b$$

Formulate as a Supervised Learning Problem

We are given labelled examples (the training set):
Inputs: $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ - Height in feet.
Targets: $\{t^{(1)}, t^{(2)}, \ldots, t^{(N)}\}$ - Avg points scored per game.

- Choose a model $\equiv$ Make an assumption about the data's behaviour. Let's say we choose -

$$y = wx + b$$

  We call w the weight and b the bias. These are the trainable parameters.

Formulate as a Supervised Learning Problem

We are given labelled examples (the training set):

Inputs: $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ - Height in feet.

Targets: $\{t^{(1)}, t^{(2)}, \ldots, t^{(N)}\}$ - Avg points scored per game.

- Choose a model $\equiv$ Make an assumption about the data's behaviour. Let's say we choose -

  $$y = wx + b$$

  We call w the weight and b the bias. These are the trainable parameters.

- Learning: Extract knowledge from the data to learn the model.

$$\begin{matrix} \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\} \\ \{t^{(1)}, t^{(2)}, \ldots, t^{(N)}\} \end{matrix} \longrightarrow \boxed{w, b}$$

Formulate as a Supervised Learning Problem

We are given labelled examples (the training set):
Inputs: $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ - Height in feet.
Targets: $\{t^{(1)}, t^{(2)}, \ldots, t^{(N)}\}$ - Avg points scored per game.

- Choose a model $\equiv$ Make an assumption about the data's behaviour. Let's say we choose -

$$y = wx + b$$

  We call w the weight and b the bias. These are the trainable parameters.

- Learning: Extract knowledge from the data to learn the model.

$$\begin{matrix}\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\} \\ \{t^{(1)}, t^{(2)}, \ldots, t^{(N)}\}\end{matrix} \longrightarrow \boxed{w, b}$$

- Inference: Given a new x and the learned model, make a prediction y.

$$x \longrightarrow \boxed{w, b} \longrightarrow y$$

Learning

Design an objective function (or loss function) that is -

- Minimized when the model does what you want it to do.
- Easy to minimize (smooth, well-behaved).

Here we want $y = wx + b$ to be close to t, for every training case.

Learning

Design an objective function (or loss function) that is -

- Minimized when the model does what you want it to do.
- Easy to minimize (smooth, well-behaved).

Here we want $y = wx + b$ to be close to t, for every training case.
Therefore one choice could be,

$$L(w, b) = \frac{1}{2}\sum_{i=1}^{N}(wx^{(i)} + b - t^{(i)})^2$$

This is called squared loss.
Need to find w, b such that $L(w, b)$ is minimized.

Learning

$$L(w, b) = \frac{1}{2} \sum_i (wx^{(i)} + b - t^{(i)})^2$$

Learning

$$
\begin{aligned}
L(w, b) &= \frac{1}{2} \sum_i (w x^{(i)} + b - t^{(i)})^2 \\
\frac{\partial L}{\partial w} &= \sum_i (w x^{(i)} + b - t^{(i)}) x^{(i)} \\
\frac{\partial L}{\partial b} &= \sum_i w x^{(i)} + b - t^{(i)}
\end{aligned}
$$

Learning

$$
\begin{aligned}
L(w, b) &= \frac{1}{2} \sum_i (wx^{(i)} + b - t^{(i)})^2 \\
\frac{\partial L}{\partial w} &= \sum_i (wx^{(i)} + b - t^{(i)}) x^{(i)} \\
\frac{\partial L}{\partial b} &= \sum_i wx^{(i)} + b - t^{(i)}
\end{aligned}
$$

Since L is a nonnegative quadratic function in w and b, any critical point is a minimum. Therefore, we minimize L by setting

$$
\frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial b} = 0.
$$

Learning

$$w\left(\sum_i x^{(i)} \cdot x^{(i)}\right) + b\left(\sum_i x^{(i)}\right) - \left(\sum_i t^{(i)} x^{(i)}\right) = 0$$

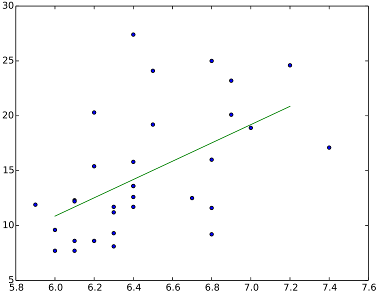
$$w\left(\sum_i x^{(i)}\right) + bN - \left(\sum_i t^{(i)}\right) = 0$$

Learning

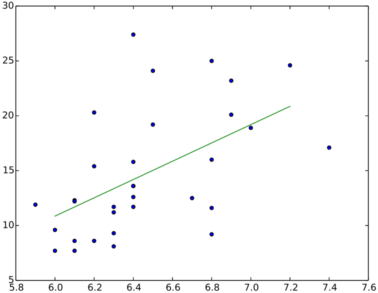
$$w\left(\sum_i x^{(i)} \cdot x^{(i)}\right) + b\left(\sum_i x^{(i)}\right) - \left(\sum_i t^{(i)} x^{(i)}\right) = 0$$
$$w\left(\sum_i x^{(i)}\right) + bN - \left(\sum_i t^{(i)}\right) = 0$$

Now we have 2 linear equations and 2 unknowns w and b. Solve!

Inference



Inference



To make a prediction about a new player, just use $y = wx + b$.

Multi-variable Linear Regression

Multi-variable : Instead of $x \in \mathbb{R}$, we have $\mathbf{x} = (x_1, x_2, \ldots, x_M) \in \mathbb{R}^M$.

Multi-variable Linear Regression

Multi-variable : Instead of $x \in \mathbb{R}$, we have $\mathbf{x} = (x_1, x_2, \ldots, x_M) \in \mathbb{R}^M$.
For example,

Height in feet (x_1)	Weight in pounds (x_2)	Avg Points Scored Per Game (t)
6.8	225	9.2
6.3	180	11.7
6.4	190	15.8
6.2	180	8.6
$\vdots$	$\vdots$	$\vdots$

Multi-variable Linear Regression

Multi-variable : Instead of $x \in \mathbb{R}$, we have $\mathbf{x} = (x_1, x_2, \ldots, x_M) \in \mathbb{R}^M$.

For example,

Height in feet (x_1)	Weight in pounds (x_2)	Avg Points Scored Per Game (t)
6.8	225	9.2
6.3	180	11.7
6.4	190	15.8
6.2	180	8.6
$\vdots$	$\vdots$	$\vdots$

Choose a model -

$$y = w_1x_1 + w_2x_2 + \ldots + w_Mx_M + b = \mathbf{w}^\top \mathbf{x} + b$$

Parameters to be learned : $\mathbf{w}$, b.

Multi-variable Linear Regression

Multi-variable : Instead of $x \in \mathbb{R}$, we have $\mathbf{x} = (x_1, x_2, \ldots, x_M) \in \mathbb{R}^M$.
For example,

Height in feet (x_1)	Weight in pounds (x_2)	Avg Points Scored Per Game (t)
6.8	225	9.2
6.3	180	11.7
6.4	190	15.8
6.2	180	8.6
$\vdots$	$\vdots$	$\vdots$

Choose a model -

$$y = w_1 x_1 + w_2 x_2 + \ldots + w_M x_M + b = \mathbf{w}^\top \mathbf{x} + b$$

Parameters to be learned : $\mathbf{w}$, b.
Objective function -

$$L(\mathbf{w}, b) = \sum_{i=1}^{N} (\mathbf{w}^\top \mathbf{x}^{(i)} + b - t^{(i)})^2$$

Multi-variable Linear Regression

We can use more general basis functions (also called "features").

$$y = w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \ldots + w_M\phi_M(\mathbf{x}) = \mathbf{w}^\top\Phi(\mathbf{x})$$

Parameters to be learned : $\mathbf{w}$.

Multi-variable Linear Regression

We can use more general basis functions (also called "features").

$$y = w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \ldots + w_M\phi_M(\mathbf{x}) = \mathbf{w}^\top \Phi(\mathbf{x})$$

Parameters to be learned : $\mathbf{w}$.
For example, 1-D Polynomial fitting

$$\begin{aligned}
\phi_0(x) &= 1 \\
\phi_1(x) &= x \\
\phi_2(x) &= x^2 \\
\phi_3(x) &= x^3 \\
\vdots &= \vdots \\
\phi_M(x) &= x^M
\end{aligned}$$

$$y = \overbrace{w_0\phi_0(x)}^{=bias} + w_1\phi_1(x) + w_2\phi_2(x) + \ldots + w_M\phi_M(x) = \mathbf{w}^\top \Phi(x)$$

Multi-variable Linear Regression

We can use more general basis functions (also called "features").

$$y = w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \ldots + w_M\phi_M(\mathbf{x}) = \mathbf{w}^\top\Phi(\mathbf{x})$$

Parameters to be learned : $\mathbf{w}$.
For example, 1-D Polynomial fitting

$$\begin{aligned}
\phi_0(x) &= 1 \\
\phi_1(x) &= x \\
\phi_2(x) &= x^2 \\
\phi_3(x) &= x^3 \\
\vdots &= \vdots \\
\phi_M(x) &= x^M
\end{aligned}$$

$$y = \overbrace{w_0\phi_0(x)}^{=bias} + w_1\phi_1(x) + w_2\phi_2(x) + \ldots + w_M\phi_M(x) = \mathbf{w}^\top\Phi(x)$$

Note : Linear regression means linear in parameters $\mathbf{w}$, not linear in $\mathbf{x}$.

Learning Multi-variable Linear Regression

Feature matrix:

$$\mathbf{\Phi} = \begin{bmatrix} \Phi(x^{(1)})^\top \\ \Phi(x^{(2)})^\top \\ \vdots \\ \Phi(x^{(N)})^\top \end{bmatrix}$$

Vector of predictions:

$$\mathbf{\Phi w} = \begin{bmatrix} \mathbf{w}^T \Phi(x^{(1)}) \\ \mathbf{w}^T \Phi(x^{(2)}) \\ \vdots \\ \mathbf{w}^T \Phi(x^{(N)}) \end{bmatrix}$$

Objective function

$$\begin{aligned} L(\mathbf{w}) &= \frac{1}{2} \sum_i (\mathbf{w}^T \Phi(x^{(i)}) - t^{(i)})^2 \\ &= \frac{1}{2} ||\mathbf{\Phi w} - \mathbf{t}||^2 \end{aligned}$$

Learning Multi-variable Linear Regression

Optimum occurs where

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \boldsymbol{\Phi}^\top (\boldsymbol{\Phi}\mathbf{w} - \mathbf{t}) = 0$$

Therefore,

$$\boldsymbol{\Phi}^\top \boldsymbol{\Phi}\mathbf{w} - \boldsymbol{\Phi}^\top \mathbf{t} = 0$$

$$\mathbf{w} = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^\top \mathbf{t}$$

Learning Multi-variable Linear Regression

Optimum occurs where

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \mathbf{\Phi}^\top (\mathbf{\Phi}\mathbf{w} - \mathbf{t}) = 0$$

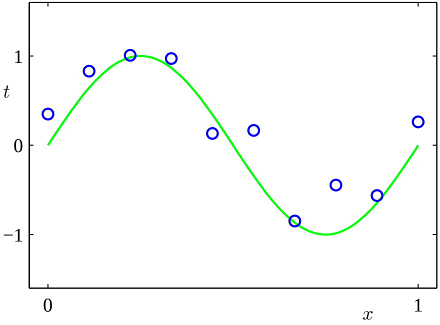
Therefore,

$$\mathbf{\Phi}^\top \mathbf{\Phi}\mathbf{w} - \mathbf{\Phi}^\top \mathbf{t} = 0$$

$$\mathbf{w} = \left(\mathbf{\Phi}^\top \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top \mathbf{t}$$

Question : When will $\mathbf{\Phi}^\top \mathbf{\Phi}$ be invertible ?

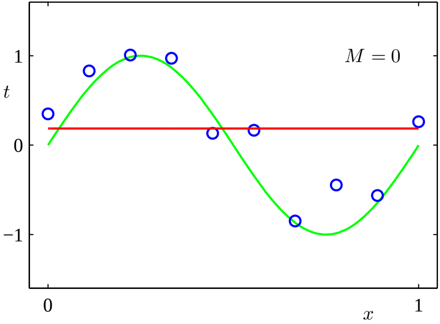
Fitting polynomials



-Pattern Recognition and Machine Learning, Christopher Bishop.

Fitting polynomials

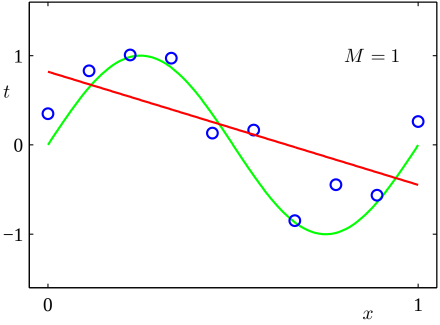
$$y = w_0$$



-Pattern Recognition and Machine Learning, Christopher Bishop.

Fitting polynomials

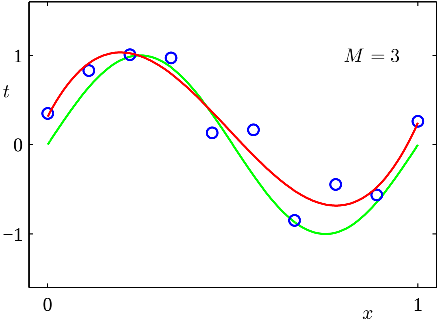
$$y = w_0 + w_1 x$$



-Pattern Recognition and Machine Learning, Christopher Bishop.

Fitting polynomials

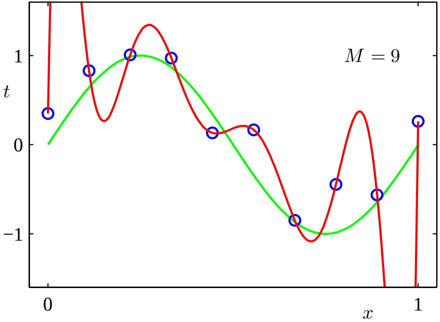
$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$



-Pattern Recognition and Machine Learning, Christopher Bishop.

Fitting polynomials

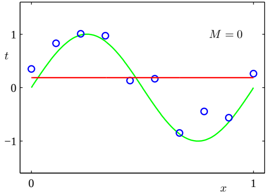
$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_9 x^9$$



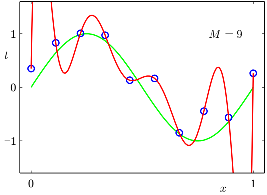
-Pattern Recognition and Machine Learning, Christopher Bishop.

Model selection

Underfitting : The model is too simple - does not fit the data.

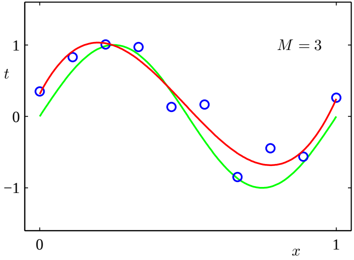


Overfitting : The model is too complex - fits perfectly, does not generalize.



Model selection

Need to select a model which is neither too simple, nor too complex.



Later in this course, we will see talk more about controlling model complexity.

Next class

- Another machine learning model, an early neural net : Perceptron.

- Frank Rosenblatt, with the image sensor (left) of the Mark I Perceptron40

Reminder - Do the quizzes for video lectures A and B by 11.59pm next Monday.