Curve Fitting
Conceptual Expansion

Dr. Kalidas Y.

In this lecture you will learn about some of the key ideas in conceptual expansion of the curve fitting

Fitting a Line Passing Through Origin

- $y = m\,x$
- $L(m) = \sum_{i=1}^{i=N}(y_i - m\,x_i)^2$
- $X = \begin{bmatrix} x_1 \\ \cdots \\ x_N \end{bmatrix}_{N\times 1}$, $Y = \begin{bmatrix} y_1 \\ \cdots \\ y_N \end{bmatrix}_{N\times 1}$, $W = [m\,]_{1\times 1}$
- $L([m]) = (XW - Y)^T(XW - Y)$
- $\nabla L = \left[\frac{\partial L}{\partial m}\right]$ //It's a function
- $W_{(new)} = W_{(old)} - \nabla L|_{W=W_{(old)}}$

Squared error type loss function

Fitting a Line – slope and intercept

- $y = m\,x + c$
- $L(m) = \sum_{i=1}^{i=N}(y_i - (m\,x_i + c))^2$
- $X = \begin{bmatrix} x_1 & 1 \\ \dots & \\ x_N & 1 \end{bmatrix}_{N\times 2}$, $Y = \begin{bmatrix} y_1 \\ \dots \\ y_N \end{bmatrix}_{N\times 1}$,

  $W = \begin{bmatrix} m \\ c \end{bmatrix}_{2\times 1}$
- $L\left(\begin{bmatrix} m \\ c \end{bmatrix}\right) = (XW - Y)^T(XW - Y)$
- $\nabla L = \begin{bmatrix} \frac{\partial L}{\partial m} \\ \frac{\partial L}{\partial c} \end{bmatrix}$ //It's a function
- $W_{(new)} = W_{(old)} - \nabla L|_{W=W_{(old)}}$

Squared error type loss function

Fitting a Parabola?

- $y = a\,x^2 + b\,x + c$
- $L(a, b, c) = \sum_{i=1}^{i=N} \big(y_i - (a\,x_i^2 +$

Squared error type loss function

Fitting a Cubic curve?

- $y = a\,x^3 + b\,x^2 + c\,x + d$
- $L(m) = \sum_{i=1}^{i=N} \big(y_i - (a\,x_i^3 + b\,x_i^2 +$

Squared error type loss function

Fitting a Degree-K polynomial?

- $y = a_k\, x^k + \cdots + a_0$
- $L(a_k, \ldots, a_0) = \sum_{i=1}^{i=N} \left(y_i - \sum_{j=0}^{k} a_j\, x^j\right)^2$
- $X = \begin{bmatrix} x_1^k \ldots x_1^2 & x_1^1 & 1 \\ & \ldots & \\ x_N^k \ldots x_N^2 & x_N^1 & 1 \end{bmatrix}_{N\times(k+1)}$,
- $Y = \begin{bmatrix} y_1 \\ \ldots \\ y_N \end{bmatrix}_{N\times 1}$,
- $W = \begin{bmatrix} a_k \\ \ldots \\ a_0 \end{bmatrix}_{(k+1)\times 1}$
- $f(X) = X \times W$
- $L(W) = (XW - Y)^T (XW - Y)$
- $\nabla L = \begin{bmatrix} \frac{\partial L}{\partial a_k} \\ \ldots \\ \frac{\partial L}{\partial a_0} \end{bmatrix}$ //It's a function
- $W_{(new)} = W_{(old)} - \eta \nabla L|_{W=W_{(old)}}$

Squared error type loss function

Steepest Descent for Multi Variate Loss function

This is a ***step size***

a.k.a ***learning rate***

$$W_{(new)} = W_{(old)} - \eta \nabla L(W) \Big|_{W=W_{(old)}}$$

This is a ***function***

specified point

of interest

Function computed ***at***

Steepest Descent for Multi Variate for Squared Error Loss function

$$\boldsymbol{W}_{(new)} = \boldsymbol{W}_{(old)} - \eta \boldsymbol{X}^{T}(\boldsymbol{X}\boldsymbol{W} - \boldsymbol{Y})$$

Multi Variate

Squared error type loss function

What is Linear About

PARABOLA Fitting??????

What is Linear About

Parabola Fitting??????

AMBIGUOUS TERMINOLOGY NEEDS SOME CLARIFICATION

Expand the Concept of Curve Fitting

Data, Label, Parameter – Simple supervised case

Data

Param

Label

$$X \times W = Y$$

Left side

Pipe Line

Right side

Fitting a Degree-K polynomial? → ANY X

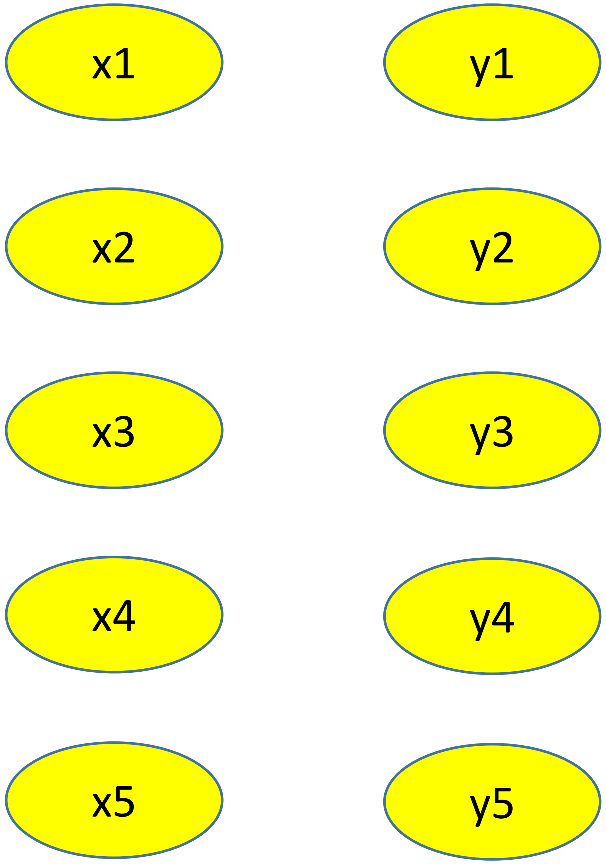
- $y = a_k\, x^k + \cdots + a_0$
- $L(m) = \sum_{i=1}^{i=N} \left(y_i - \sum_{j=0}^{k} a_j\, x^j\right)^2$
- $X = \begin{bmatrix} x_1^k \ldots x_1^2 & x_1^1 & 1 \\ & \ldots & \\ x_N^k \ldots x_N^2 & x_N^1 & 1 \end{bmatrix}_{N\times(k+1)}$,
- $Y = \begin{bmatrix} y_1 \\ \ldots \\ y_N \end{bmatrix}_{N\times 1}$,
- $W = \begin{bmatrix} a_0 \\ \ldots \\ a_k \end{bmatrix}_{(k+1)\times 1}$
- $L(W) = (XW - Y)^T(XW - Y)$
- $\nabla L = \begin{bmatrix} \frac{\partial L}{\partial a_0} \\ \ldots \\ \frac{\partial L}{\partial a_k} \end{bmatrix}$ //It's a function
- $W_{(new)} = W_{(old)} - \nabla L|_{W=W_{(old)}}$

Squared error type loss function

any.. X Formulation

$$X \times W = Y$$

Representation



Mapping

x_i vector to y_i vector

List notation

$Y = f(X)$

f() is called model

Linear Model

- $X = \begin{bmatrix} x_{1,1}, \dots, x_{1,k} \\ \dots \\ x_{N,1}, \dots, x_{N,k} \end{bmatrix}_{N\times k}$,
- $Y = \begin{bmatrix} y_1 \\ \dots \\ y_N \end{bmatrix}_{N\times 1}$,
- $W = \begin{bmatrix} w_1 \\ \dots \\ w_k \end{bmatrix}_{k\times 1}$

- $L(W) = (XW - Y)^T(XW - Y)$
- $\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \dots \\ \frac{\partial L}{\partial w_k} \end{bmatrix}$ //It's a function
- $W_{(new)} = W_{(old)} - \nabla L|_{W=W_{(old)}}$

$$f(x) = x^T W$$

Input point x is a vector
k x 1 matrix

This is the Linear Part of the "Linear" regression phrase

Squared error type loss function

Linear Model

- $X = \begin{bmatrix} x_{1,1}, \dots, x_{1,k} \\ \dots \\ x_{N,1}, \dots, x_{N,k} \end{bmatrix}_{N \times k}$,
- $Y = \begin{bmatrix} y_1 \\ \dots \\ y_N \end{bmatrix}_{N \times 1}$,
- $W = \begin{bmatrix} w_1 \\ \dots \\ w_k \end{bmatrix}_{k \times 1}$

- $L(W) = (XW - Y)^T (XW - Y)$
- $\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \dots \\ \frac{\partial L}{\partial w_k} \end{bmatrix}$ //It's a function
- $W_{(new)} = W_{(old)} - \nabla L|_{W=W_{(old)}}$

$$L(W) = \sum_{i=1}^{i=N} \left(y_i - \left(\sum_{j=1}^{j=k} w_j \, x_{i,j} \right) \right)^2$$

Squared error type loss function

Linear Model Multi-Variate

- $X_{N\times k} = \begin{bmatrix} x_{1,1}, \dots, x_{1,k} \\ \dots \\ x_{N,1}, \dots, x_{N,k} \end{bmatrix}_{N\times k}$,
- $Y_{N\times m} = \begin{bmatrix} y_{11}, y_{1,2} \dots y_{1,m} \\ \dots \\ y_{N1}, y_{N,2} \dots y_{N,m} \end{bmatrix}_{N\times m}$,
- $W_{k\times m} = \begin{bmatrix} w_{1,1} \dots w_{1,m} \\ \dots \\ w_{k,1} \dots w_{k,m} \end{bmatrix}_{k\times m}$
- $L(W) = g((XW - Y)^T(XW - Y))$
- where, $g(A_{m\times m}) \rightarrow scalar$
- $\nabla L_{k\times m} = \begin{bmatrix} \frac{\partial L}{\partial w_{1,1}}, \dots, \frac{\partial L}{\partial w_{1,m}} \\ \dots \\ \frac{\partial L}{\partial w_{k,1}}, \dots, \frac{\partial L}{\partial w_{k,m}} \end{bmatrix}$ //It's a function
- $W_{(new)} = W_{(old)} - \eta \nabla L|_{W=W_{(old)}}$

Brain storm applications…

Depends on X and Y

- For example, Audio recording → file size prediction
- For example, Image → Cat-like-picture-*ness* prediction
  [we will see classification formulation later]
- For example, Video → A given scene-like-*ness* prediction
  [we will see classification formulation later]
- For example, Text → A given meaning-like-*ness* prediction
  [we will see classification formulation later]
- For example, Measurements → A given *concept*-like-*ness* prediction
- *For example, Housing colony data → House price prediction (California house price prediction, Boston house price prediction)*

Depends on X and Y (multi variate)

- **The whole of the deep neural networks is an extension of this concept!**
- ***Map*** one image to another
- Given a set of images, ***fill-in the gap*** regions of an image
- Given a set of audio clips, ***fill in the gaps*** of an audio clip
- Given a sentence in one language, ***translate*** a sentence in another language
- Given an image, ***map*** it to text sentence
- Given a text label, ***map*** it to generate an image
- *Hundreds of applications… so many use cases, so many domains…*

…people started to think that it is the end of the world? But not!! {Refer to what is and what is not ml slides}

- ***Given a vector map it to Another vector***