Note:

- HW#5 submission closed now.
- All homework are completed now ☺

Note:

- Continue working on your lit. survey project.
- Presentation format: www.pechakucha.com. More info later.
- Project report due **in three weeks**
  - Read the assignment thoroughly
  - Late policy will apply.
- Project presentation **in three weeks**
  - Submit your slides two days prior (**5/11, 5pm**) by **email**
  - **Read the assignments again to refresh your mem.**
  - **More details for presentation will be emailed soon**
  - **Presentation in alphabetical order**

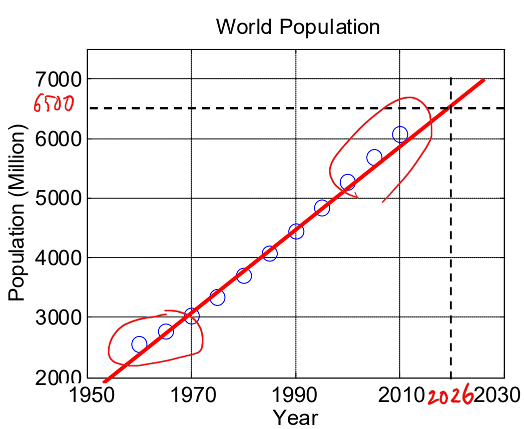
Classification Supervised ML

Regression & Learning

CSC 872
Pattern Analysis and Machine Intelligence

What is Regression ?

Statistics101



What is the population in year 2026

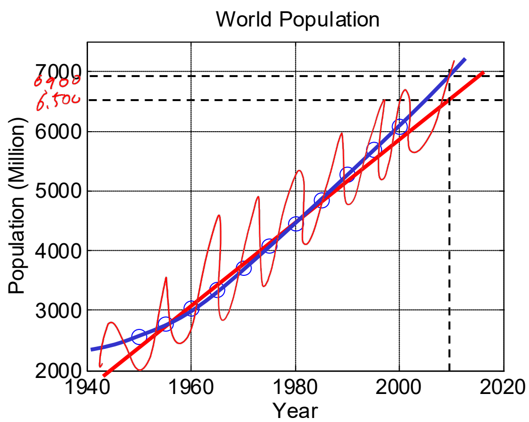
1. Fit a line

2. Find our prediction

- Regression is a statistical analysis to find a function representing input-output relation from data samples

http://www.populationmedia.org/issues/popgrowth_data.html

What is Regression ?



What is the population in year 2026?

1. Fit a curve

2. Find our prediction

- We must choose appropriate function form to do regression

http://www.populationmedia.org/issues/popgrowth_data.html

In Nutshell

- Goal: estimate input-output relation from data
- For: prediction/forecasting/modeling
- You need to
  1) *Pick a form of parametric function*
     - **Line**:
     - **Polynomial Curves**:
     - **General Curves**:
     - **General Functions**:

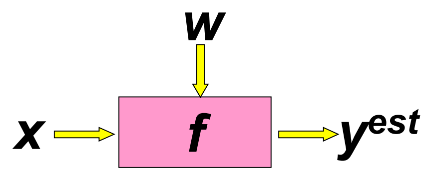
     Simple

     Model Selection

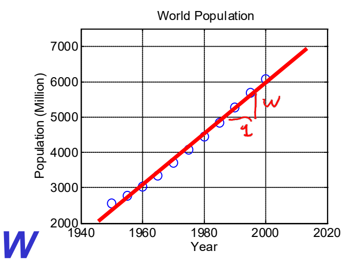
     Complex
  2) *Fit the function to the data*
     - **Maximum likelihood estimation (MLE) is the foundation**

Learning Machine Interpretation

- Learning Machine: ***y = f(x)***



- **Input**: independent variable
  - E.g., ***x*** = year
- **Output**: dependent variable
  - E.g., ***y*** = population
- **Function**: parameterized by ***W***
  - E.g., ***f***(x,w) = wx: line

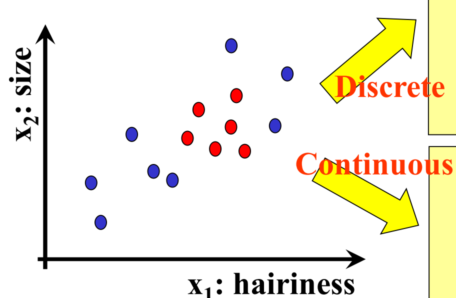


Training Data X (Supervised ML)

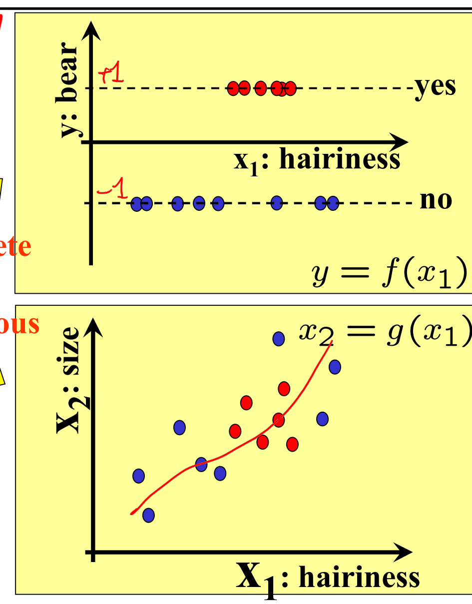
- Supervised learning $x_i \in \mathbb{R}^M$ $y_i \in \mathbb{R}$

  $y_i = f(x_i, w)$

$$\boldsymbol{X} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N=14}$$



$$y_i = \begin{cases} +1 & \text{if bear} \\ -1 & \text{otherwise} \end{cases}$$



Regression Types

Depends on the form of parameterized functions

- – Linear Regression (line/plane/hyperplane)
- – Polynomial Regression (polynomial curve)
- – Non-linear Regression (general curve)
- – Radial-Basis Function Regression (basis sum)
- – Piecewise Linear Regression (line segments)
- – Non-parametric Regression (KDE)
- – Robust Regression (robust estimation!!!)

Regression Types

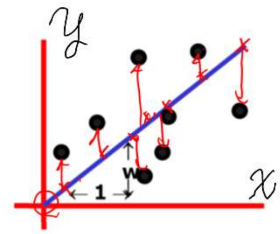
Depends on the form of parameterized functions

- – Linear Regression (line/plane/hyperplane)
- – Polynomial Regression (polynomial curve)
- – Non-linear Regression (general curve)
- – Radial-Basis Function Regression (basis sum)
- – Piecewise Linear Regression (line segments)
- – Non-parametric Regression (KDE)
- – Robust Regression (robust estimation!!!)

Linear Regression

- **Simplest one parameter case**

  $y \in \mathbb{R}, x \in \mathbb{R}, w \in \mathbb{R}.$



- Data is formed by: $\boldsymbol{y = wx + noise}$
  - Unknown scalar $\boldsymbol{w}$
  - Noise is independent **random variable**
  - Noise is normally-distributed with zero mean & σ^2

    noise model $\longrightarrow \mathcal{N}(0, \sigma^2)$
- Output $\boldsymbol{y}$ is then also a random variable $\mathcal{N}(wx, \sigma^2)$

  with $\boldsymbol{P(y|x,w) = Normal(\,mean\ \text{“}wx\text{”}, variance\ \text{“}\sigma^2\text{”})}$
- Given data: N $\boldsymbol{i.i.d.}$ evidences $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
- Regression Problem: *Find **w** from data such that ….*

Bayesian Linear Regression

- *Find w from data such that **it maximizes the posterior distribution:***

$$\boldsymbol{P(\,w \mid (x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\,)}$$

- Infer $\boldsymbol{P(w|data)}$ from data likelihood $\boldsymbol{P(y|x,w)}$ using Bayes rules ! $P(x)$
  - Conjugate prior etc, A bit complicated so…

Maximum-Likelihood Estimate

- *Find w from data such that* ***it maximizes the data likelihood function:***

$$P(\, \boldsymbol{y} \mid \boldsymbol{x_1}, \boldsymbol{x_2}, .., \boldsymbol{xN}, \boldsymbol{w} \,) \; = \; \boldsymbol{N}(\boldsymbol{y};\; \boldsymbol{wx}, \sigma^2)$$

- As usual, Let's do some algebra to simplify

Algebra Joy: You know this by now

- For what ***w*** is this data most likely to have happened?
- For what ***w***, is

$$P(y_1, .., y_N | x_1, .., x_N, w) \text{ maximized?}$$

- For what ***w***, is

$$\prod_{i=1}^{N} P(y_i|x_i, w) \text{ maximized?}$$

i.i.d.

- For what ***w***, is

N(wx, σ²)

$$\prod_{i=1}^{N} \exp\left(-\frac{1}{2}\frac{(y_i - wx_i)^2}{\sigma^2}\right) \text{ maximized?}$$

log

- For what ***w***, is

$$\sum_{i=1}^{N} -\frac{1}{2}\frac{(y_i - wx_i)^2}{\sigma^2} \quad \text{maximized?}$$

This is known as Least Squares method

- For what ***w***, is

algebra

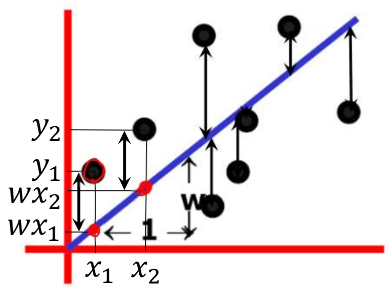
$$\boxed{\sum_{i=1}^{N} (y_i - wx_i)^2 \quad \text{minimized?}}$$

Least Squares Method

$y = wx + \text{noise}$
$N(0, \sigma^2)$

- MLE of ***w*** is one that minimizes the sum-of-squares of residuals (errors) Least Squares

$$E(w) = \sum_i (y_i - wx_i)^2$$



$$\sum \left\{ \begin{matrix} (y_1 - wx_1)^2 \\ (y_2 - wx_2)^2 \\ \vdots \\ (y_i - wx_i)^2 \\ \vdots \\ (y_N - wx_N)^2 \end{matrix} \right\}$$

Really a quadratic optimization

$E(w) = aw^2 - bw + c$

- MLE of ***w*** is one that minimizes the sum-of-squares of residuals (errors)

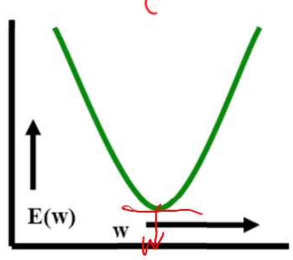
$\frac{\partial E(w)}{\partial w} = 2aw - b = 0$
$\rightarrow w = \frac{b}{2a}$

$$E(w) = \sum_i (y_i - wx_i)^2 = \sum_i (x_i^2 w^2 - 2x_i y_i w + y_i^2)$$

$$= \underbrace{(\sum_i x_i^2)}_{a} w^2 - \underbrace{\left(2 \sum_i x_i y_i\right)}_{b} w + \underbrace{(\sum_i y_i^2)}_{c}$$

- We want to minimize a quadratic function of ***w***

$$\frac{\partial E(w)}{\partial w} = 0$$



STOP: We have a closed-form solution!

- For linear regression w/ normal-distributed noise
- **MLE = Least Squares !!!**

$$w^{mle} \doteq argmin_w E(w) = argmin_w \sum_i (y_i - wx_i)^2$$

$$\Leftrightarrow \frac{\partial E(w)}{\partial w} = 0$$

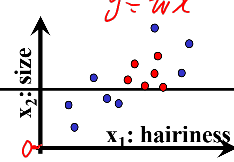
$$w = \frac{\overset{b}{\sum_i x_i y_i}}{a \sum_i x_i^2}$$

r: Pearson correlation coefficient
σ_x: standard deviation of $\{x_i\}$
σ_y: standard deviation of $\{y_i\}$

$$y = wx = \frac{\sum_i x_i y_i}{\sum_i x_i^2} x = \frac{\sigma_y}{\sigma_x} r x$$

Multivariate Case? $x_i = \begin{pmatrix} x_1 \\ x_i \\ \vdots \\ x_M \end{pmatrix}$

$y = wx$



- What if input $\boldsymbol{x}$ is a vector $(x_1,..,x_M)$^T?
- Model is $y = \boldsymbol{w}^t \boldsymbol{x} + \epsilon$
  $= w_1 x_1 + w_2 x_2 + \cdots + w_m x_m + \cdots w_M x_M + \epsilon$
- Given data: $(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),..,(\boldsymbol{x}_N,y_N)$ $\boxed{Y = Xw + \varepsilon}$ $w = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix}$

a set of Linear Eq.

$$\boldsymbol{X} = \begin{bmatrix} x_1^t \\ x_2^t \\ \vdots \\ x_N^t \end{bmatrix} = N \underbrace{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix}}_{M} \quad \boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$w = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$

- MLE of $\boldsymbol{w}$ is given by
  $\boldsymbol{w}^{mle} = (\boldsymbol{X}^t \boldsymbol{X})^{-1} \boldsymbol{X}^t \boldsymbol{Y}$ **(Pseudo Inverse)**

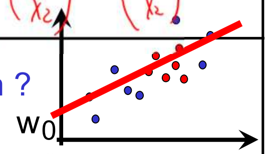
$$y = \boldsymbol{w}^t \boldsymbol{x} = ((\boldsymbol{X}^t \boldsymbol{X})^{-1} \boldsymbol{X}^t \boldsymbol{Y})^t \boldsymbol{x}$$

$Y = wX$
$X^{-1} Y = X^{-1} w X = w$
$\rightarrow w = X^{-1} Y$

Constant Term?

$y = wx$ $\quad w = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$ $\quad x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$ $\quad z$

- What if the line does not intersect the origin ?



X_1	X_2	Y
2	4	16
3	4	17
5	5	20

$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ $\quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$$Y = w_1X_1 + w_2X_2 + \epsilon$$
$$= w^t x + \varepsilon$$

X_0	X_1	X_2	Y
1	2	4	16
1	3	4	17
1	5	5	20

$w^t x + w_0 + \varepsilon$

$$Y = w_0 + w_1X_1 + w_2X_2 + \epsilon$$
$$= \underline{w_0X_0 + w_1X_1 + w_2X_2} + \epsilon$$
$$= w^t \begin{pmatrix} 1 \\ x \end{pmatrix} + \varepsilon$$

- Model is

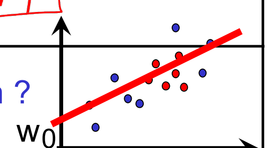
$$y = \boldsymbol{w^t z} + \epsilon, \qquad \boxed{\boldsymbol{z} = (1, \boldsymbol{x}^t)^t}$$
$$= w_0 1 + w_1x_1 + w_2x_2 + \cdots + w_Mx_M + \epsilon$$

$w_0 = w_0 \times 1$

Constant Term?

Linear SVM

- What if the line does not intersect the origin ?
- Model is $y = \boldsymbol{w^t z} + \epsilon, \boldsymbol{z} = (1, \boldsymbol{x}^t)^t$



Add new column of 1s

- Given data: *($\boldsymbol{x}_1$,y_1),($\boldsymbol{x}_2$,y_2),…,($\boldsymbol{x}_N$,y_N)*

$Y = Zw + \varepsilon$

X

$$\boldsymbol{Z} = \begin{bmatrix} z_1^t \\ z_2^t \\ \vdots \\ z_N^t \end{bmatrix} = \begin{bmatrix} (1, x_1^t)^t \\ (1, x_2^t)^t \\ \vdots \\ (1, x_N^t)^t \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1M} \\ 1 & x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix} \qquad \boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

- MLE of ***w*** is given by

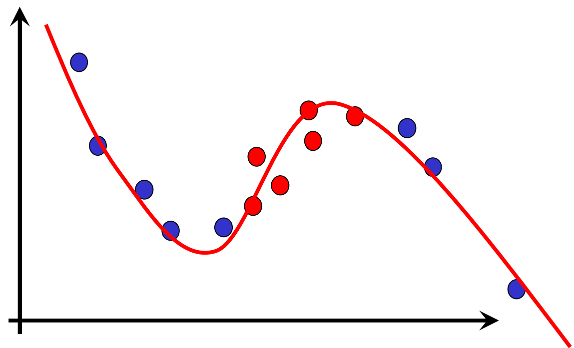
$$\boldsymbol{w} = (\boldsymbol{Z^tZ})^{-1}\boldsymbol{Z^tY}$$

Most general linear regression formula !!

$$y = \boldsymbol{w^t z} = ((\boldsymbol{Z^tZ})^{-1}\boldsymbol{Z^tY})^t \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{x} \end{bmatrix}$$

Polynomial Regression

- What if I want to fit a polynomial curve?



- You can reuse the linear formula!!!

Quadratic Regression

$y = wx + \varepsilon$
$y = w^t x + w_0 + \varepsilon$
$= w_0 + w_1 x_1 + w_2 x_2 + \varepsilon$

linear Quadratic

- Model for 2D is

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + w_5 x_2^2 + \epsilon$$
$$= \boldsymbol{w}^t \boldsymbol{z} + \epsilon, \qquad \boldsymbol{z} = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$

- Given data: $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)$, $\boldsymbol{x}_i = (x_{i1}, x_{i2})$

$Y = Zw + \varepsilon$

$$\boldsymbol{Z} = \begin{bmatrix} z_1^t \\ z_2^t \\ \vdots \\ z_N^t \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{11}^2 & x_{11}x_{12} & x_{12}^2 \\ 1 & x_{21} & x_{22} & x_{21}^2 & x_{21}x_{22} & x_{22}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & x_{N2} & x_{N1}^2 & x_{N1}x_{N2} & x_{N2}^2 \end{bmatrix} \qquad \boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

You change here

- MLE of $\boldsymbol{w}$ is given by

$$\boldsymbol{w} = (\boldsymbol{Z}^t \boldsymbol{Z})^{-1} \boldsymbol{Z}^t \boldsymbol{Y}$$

But all these are same

$$\boxed{y = \boldsymbol{w}^t \boldsymbol{z} = ((\boldsymbol{Z}^t \boldsymbol{Z})^{-1} \boldsymbol{Z}^t \boldsymbol{Y})^t \boldsymbol{z}}$$

Q^{th} degree Polynomial Regression

- Model is the same but with different $\boldsymbol{z}$ long

$$y = \boldsymbol{w}^t \boldsymbol{z} + \epsilon, \qquad \boldsymbol{z} = (1, q(\boldsymbol{x}))$$

- Given data: $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), .., (\boldsymbol{x}_N, y_N)$
- *$q(\boldsymbol{x})$: all products of powers of inputs up to Q^{th} degree*

$$\boldsymbol{Z} = \begin{bmatrix} z_1^t \\ z_2^t \\ \vdots \\ z_N^t \end{bmatrix} = \begin{bmatrix} 1 & q(\boldsymbol{x}_1) \\ 1 & q(\boldsymbol{x}_2) \\ \vdots & \vdots \\ 1 & q(\boldsymbol{x}_N) \end{bmatrix} \qquad \boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

You cannot model complicated curves

Because Z gets large very quick by increasing the complexity of the curve by increasing Q when more than 1D input

- MLE of $\boldsymbol{w}$ is given by

$$\boldsymbol{w} = (\boldsymbol{Z}^t \boldsymbol{Z})^{-1} \boldsymbol{Z}^t \boldsymbol{Y}$$

$$\boxed{y = \boldsymbol{w}^t \boldsymbol{z} = ((\boldsymbol{Z}^t \boldsymbol{Z})^{-1} \boldsymbol{Z}^t \boldsymbol{Y})^t \boldsymbol{z}}$$

Radial Basis Function Regression

- Can we generalize the idea of the Polynomial Regression?
  - Basically, you construct Z with different z with various products of inputs
  - Then, use the same pseudo inverse formula
- Let's construct $\boldsymbol{z}$ with some function $\phi(x)$ of input $\boldsymbol{x}$
- Model is $y = w_0 + w_1\phi_1(\boldsymbol{x}) + \cdots + w_K\phi_K(\boldsymbol{x}) + \epsilon$ Meron Kovel

$$= \boldsymbol{w}^t \boldsymbol{z} + \epsilon, \qquad \boldsymbol{z} = (1, B(x) = \phi_1, .., \phi_K)$$

- B(x) is called a **basis** whose linear combination gives an output
- We choose the basis to be symmetric about a center $\boldsymbol{c}$ with spread $\boldsymbol{W}$ then call it **radial basis function**

$$\phi_k(x) = \mathrm{RadialBasisFunction}\left(\frac{|x - c_k|}{W_k}\right)$$

- RBF Regression performs the linear regression with B(x) defined with the radial basis function

Non-linear Regression

- What if I want to fit more general nonlinear function *f(**x**;**w**)*?
- Let's do the same as before !
  - Assume a general model of $y = f(x; w) + \epsilon$
  - Normally-distributed independent noise
  - Likelihood ***P(Y|X,w)*** is *Normal(mean **f(x;w)**,variance σ^2)*
  - MLE of ***w*** = LS of ***w***

$$\boldsymbol{w} = \text{argmin}_w \sum_{i=1}^{N} (y_i - f(\boldsymbol{x_i}; \boldsymbol{w}))^2$$

$$\Leftrightarrow \frac{\partial}{\partial \boldsymbol{w}} \sum_i (y_i - f(\boldsymbol{x_i}; \boldsymbol{w}))^2 = 0$$

$$\Leftrightarrow \sum_i (y_i - f(\boldsymbol{x_i}; \boldsymbol{w})) \frac{\partial f(x_i; w)}{\partial w} = 0$$

Ooops! how to solve this about ***w***???
We are doomed. we are stuck here?…

Energy Optimization

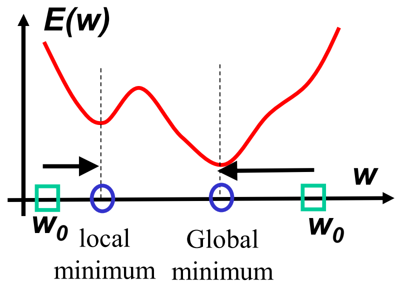
- **Recall the non-parametric modeling lecture…**
- The savior is to go "**Iterative**" to solve $w = \text{argmin}_w E(w)$
- Minimizing the Energy/Error/Cost/Potential function by
  - Define an iterative step ***move(w,E(w))***
  - Then find an initial solution $\boldsymbol{w_0}$
  - Then find a sequence $\boldsymbol{w_0, w_1, \ldots, w_m}$ by doing

$$\begin{aligned}
&\text{step1:} \quad w_{new} = w_{old} + move(w_{old}, E(w)) \\
&\text{step2:} \quad w_{old} = w_{new} \\
&\text{step3:} \quad \text{go to step1}
\end{aligned}$$

- To do this right, you need to find ***move(w,E(w))*** so that $\boldsymbol{w}_m$ converges to a local minimum of ***E(w)***

In general

- You want $\mathbf{w} = \text{argmin}_w E(\mathbf{w})$



Iterative solution can get stuck at local minimum depending on initialization

- What if I have a maximization problem ?

$$w = \text{argmax}_w E(w)$$

$$w = \text{argmin}_w - E(w)$$

You can always interchange minimization and maximization problem

How are we going to solve it?

Optimization
Optimization

- Various ways
  - Line search
  - Hill-Climbing
  - Gradient-Descent (Steepest-Descent/Ascent)
  - Conjugate-Gradient
  - Levenberg-Marquart
  - Newton's Method
  - Simulated Annealing
  - EM-algorithm
  - Mean Shift
  - More and more…

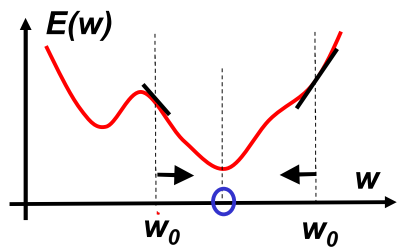
We study this during the lectures for search methods in AI

statistical modeling

Gradient-Descent

- We define the step move function as a negative of partial derivatives of the energy w.r.t. the unknown parameter W

$$w_{new} = w_{old} - \underline{\eta \frac{\partial E(w)}{\partial w}\bigg|_{w=w_{old}}} \qquad \eta > 0$$



η is a learning rate set to a small constant (e.g., 0.05)

Steepest-Descent/Ascent

- Umm, sorry but, I want to maximize really, want to go up
- **DON'T LIKE GOING DOWN!!!**
- Just flip the sign!!!

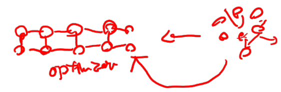
$$w_{new} = w_{old} \underline{+} \eta \frac{\partial E(w)}{\partial w}\bigg|_{w=w_{old}}$$

Gradient-Ascent

- **Steepest-Ascent**: a type of greedy iterative search algorithm we learned in our lecture on search for AI

- Umm: the same
- **Gradient-Ascent = Steepest-Ascent**

Simulated Annealing



- Are there a way to avoid getting stuck in a local minimum?
- Yes: called "simulated annealing" making it stochastic

  step1: make a **random** move $W_{old} \xrightarrow[\text{random}]{} W_{new}$ $E\uparrow$

  step2: take this move if reduces the energy

  step3: else take it with cetain acceptance probability

  step4: go to step1

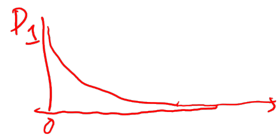
  RGB

  negative

  $$\text{Acceptance Prob.} \quad P = \exp\left(\frac{E(w_{old}) - E(w_{new})}{T_i}\right) \quad e^{-\frac{\Delta E}{T}}$$

  sampling from Maxwell–Boltzman distribution

- When temperature T_i is $\frac{\Delta E}{T}$
  - **High: random walk**
  - **Low: stochastic steepest descent**



$e^0 = 1$

- <u>Can converge to the global minimum when scheduling a gradual decreasing of the temperature (**cooling schedule**)</u>

Summary

- Regression & Learning
  - What is regression?
  - Maximum Likelihood Estimate & Least Squares Method
  - Linear regression
  - Polynomial regression
  - Radial Basis Function regression
  - Gradient-descent
  - Simulated Annealing
- Next
  - Artificial Neural Network
  - End of the LDA FP

Artificial Neural Network

- Today's lecture on ML-based regression can be directly used to understand and implement ANN!
- The note will be uploaded in the course web
- Read the note!
  - Neural Network
  - Perceptron
  - Multi-Layered Network
  - Backpropagation
  - Other Networks