Lecture 7: Regression
Modeling Social Data, Spring 2019
Columbia University

March 8, 2019

Notes from bs3118

1 Reproducibility

As discussed in the previous class, there is currently a crisis of reproducibility and replication. To counter this, there a few guidelines we should follow before we publish research.

- Read all the pertinent literature, as you may not be the first person to have this idea. Literature surveys also help in understanding our problem better
- Formulate your study
- Run a pilot to test the study
- Analyze the results of the study
- Revise your study based on this analysis
- Calculate the power of your study, to define the effect and population size needed
- Pre-regiser plans for your study online. This helps in reviewing your process, and ensures that you cannot lie about the results later on.
- Run your study
- Create a detailed and reproducible report using the makefile templates define during the pilot study
- Think critically about your results
- Disclose everything you did during the study, including data collection, preprocessing etc.

Other good practices for reproducibility include:

- Use Mechanical Turk to collect data since it is reliable and quick
- Running a pilot study first
- Creating a framework to analyse the data that can be easily reproduced based on the pilot study
- To ensure that the subjects in the study understand the questions, we should add some questions as sanity checks.
- We could also use timestamps of the data collected to remove any bots.

Also, we should remember:

Standard deviation: The variation in the data (this remains constant no matter how many samples we choose)

Standard error: This represents how far the sample mean is from the population mean. Thus, if we take more number of samples, we can reduce this.

2 Regression

Regression is used to estimate the relationship between variables. It can be used to summarize the data, predict future data, and explain relationships within the data. We need to find a balance between understanding out current data, and being able to generalize in a way that we can also predict future outcomes. The task is to find a function to do this. The notes below show the how the task is defined.

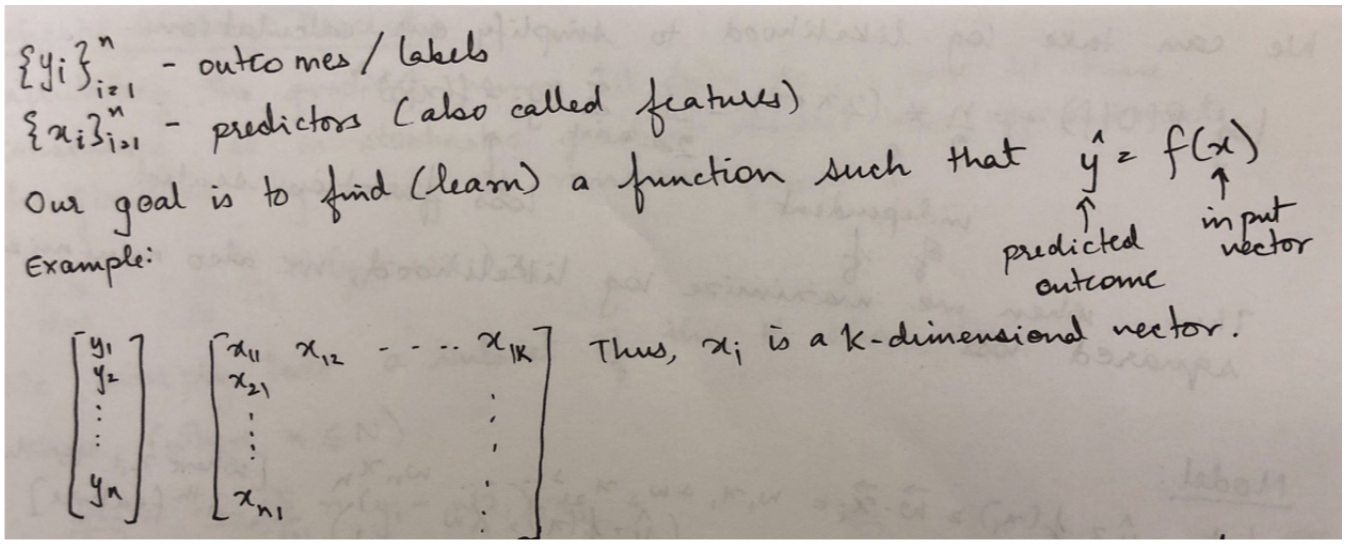


Figure 1: Defining the task

2.1 Task

2.2 Loss functions

We then define loss functions to evaluate our function. The notes below show the how the task is defined.

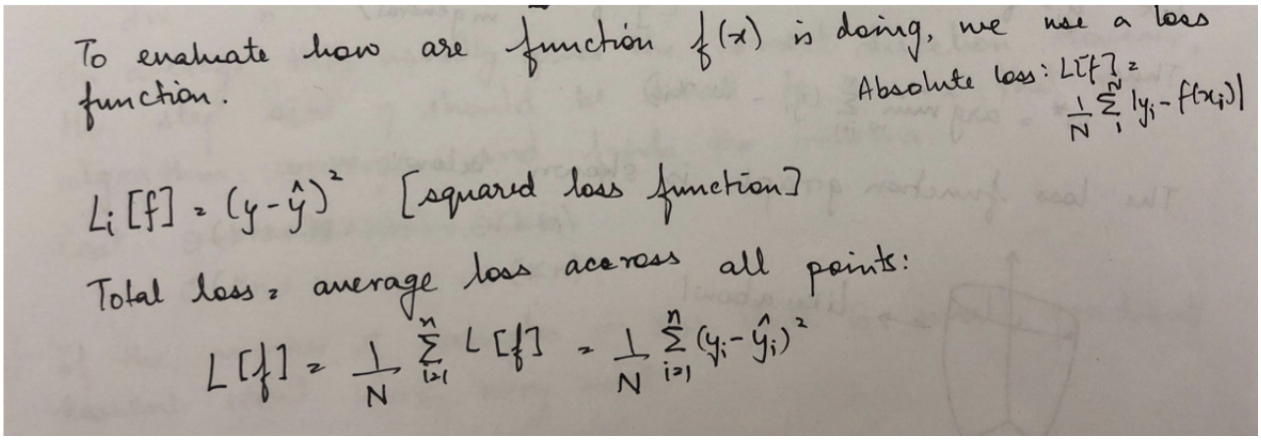


Figure 2: Define loss functions

2.3 MLE

Maximum Likelihood Estimator is used to find how to minimize loss.

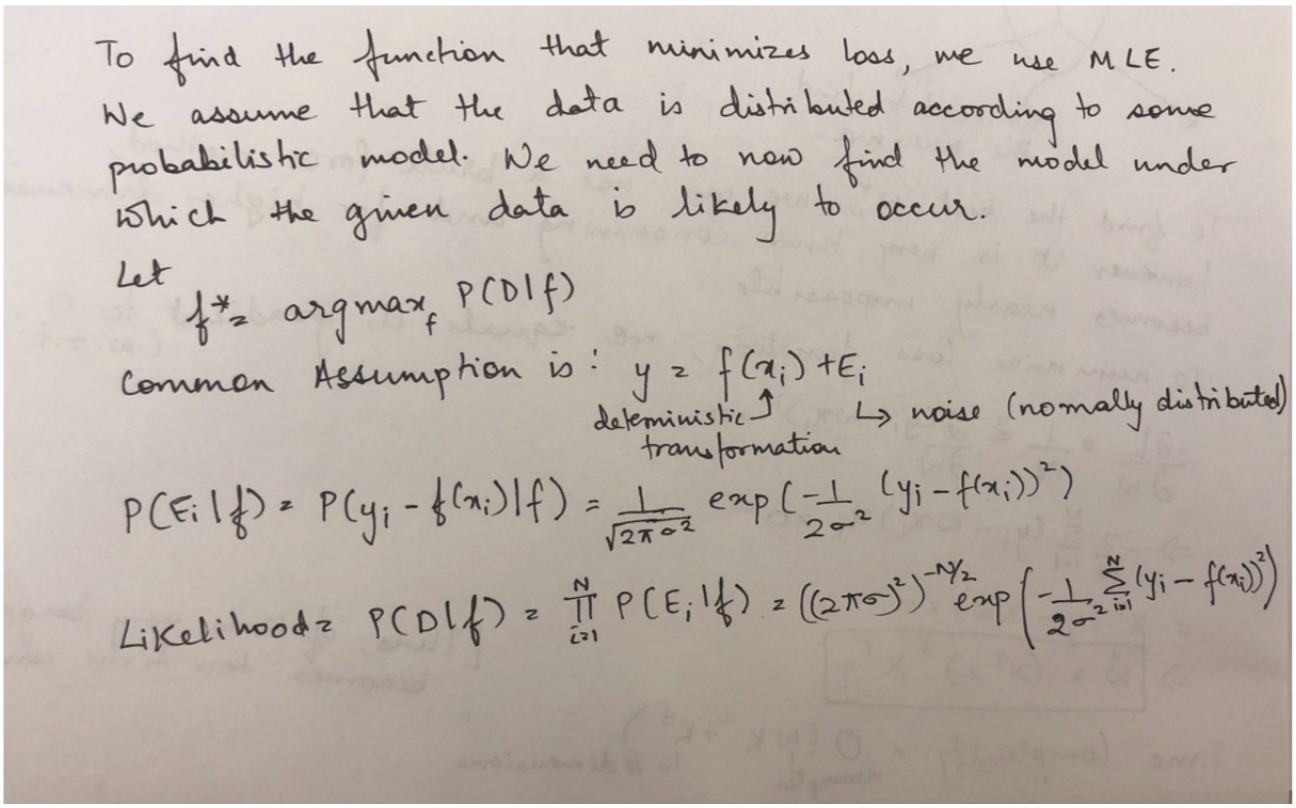


Figure 3: MLE

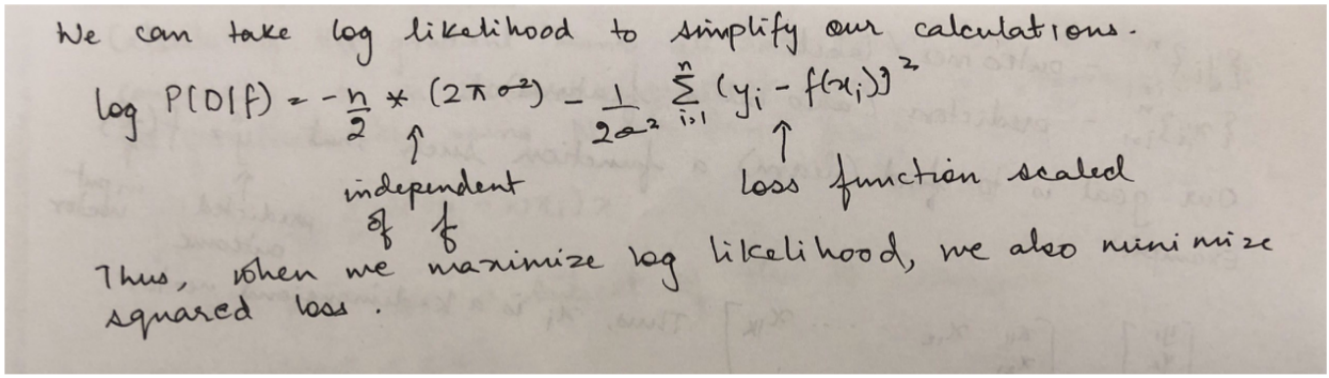


Figure 4: MLE

2.4 Finding f(x)

We then define the regression model in terms of weights. To find the best possible value of the weight vector, we can use three main methods: Normal equations (shown in Figure 5), Gradient Descent (Figure 6), and Stochastic Gradient Descent (shown in Figure 7)

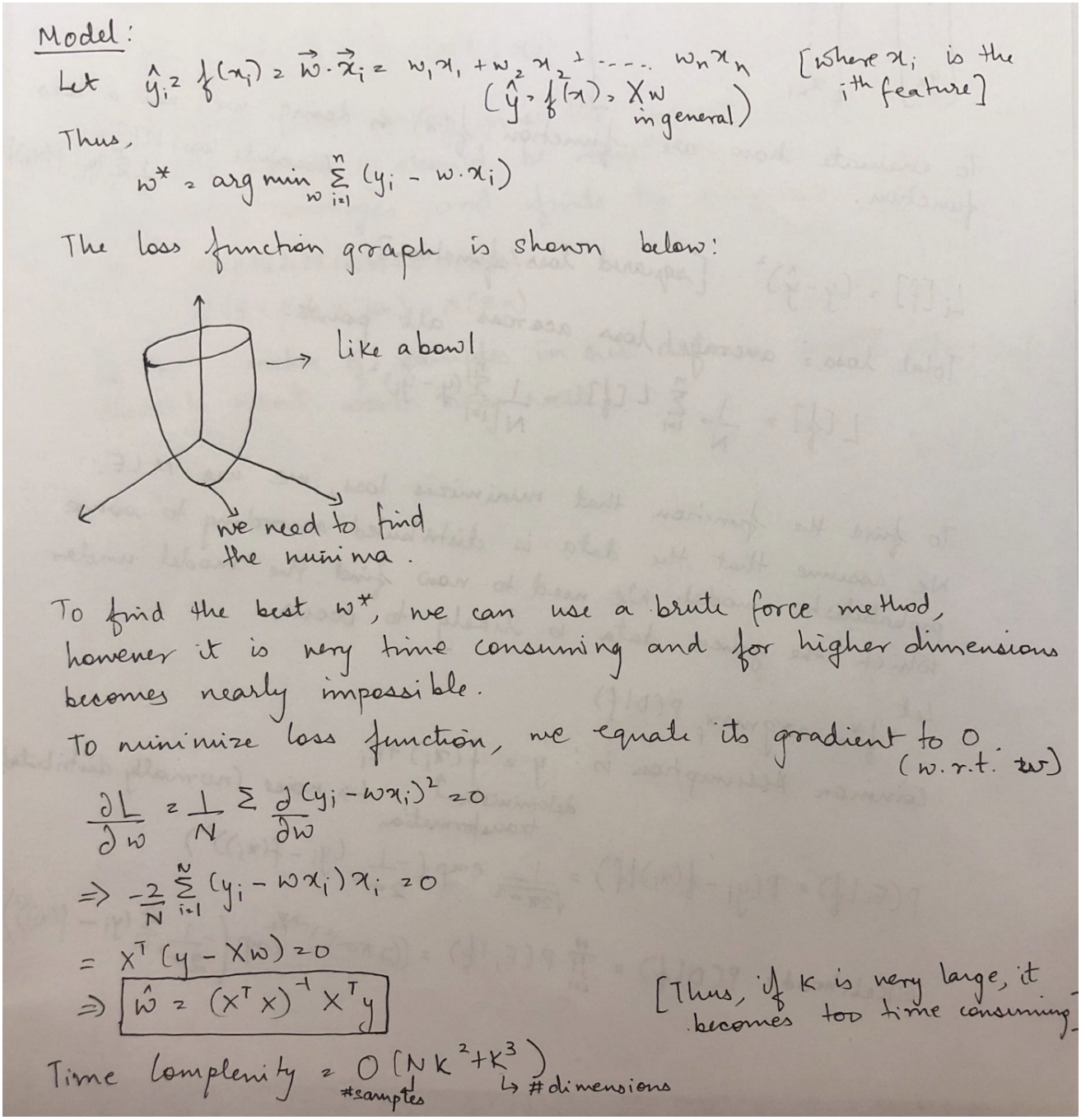


Figure 5: Defining and determining f(x)

2.5 Gradient Descent

Gradient descent can be used to estimate the w vector.

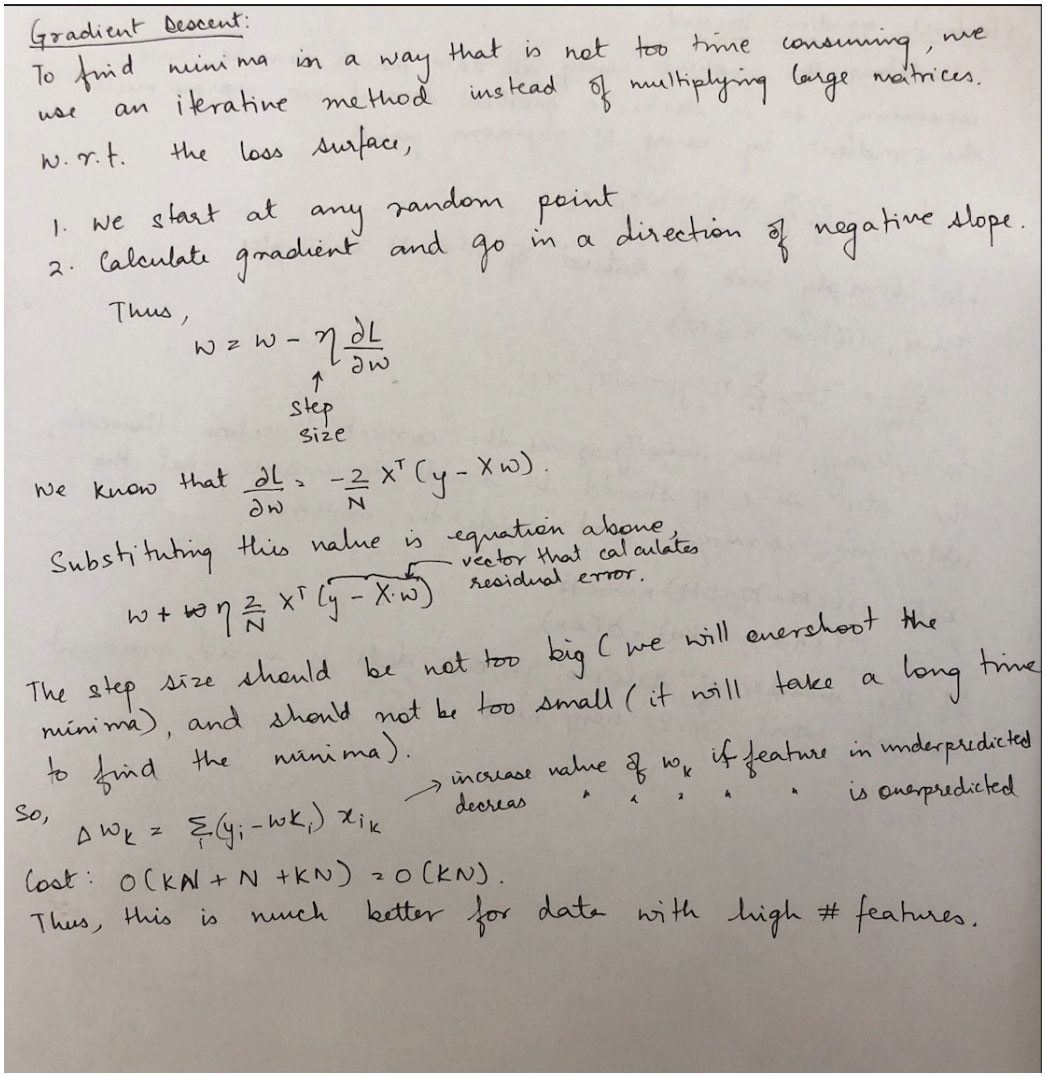


Figure 6: Gradient Descent

2.6 Stochastic gradient descent:

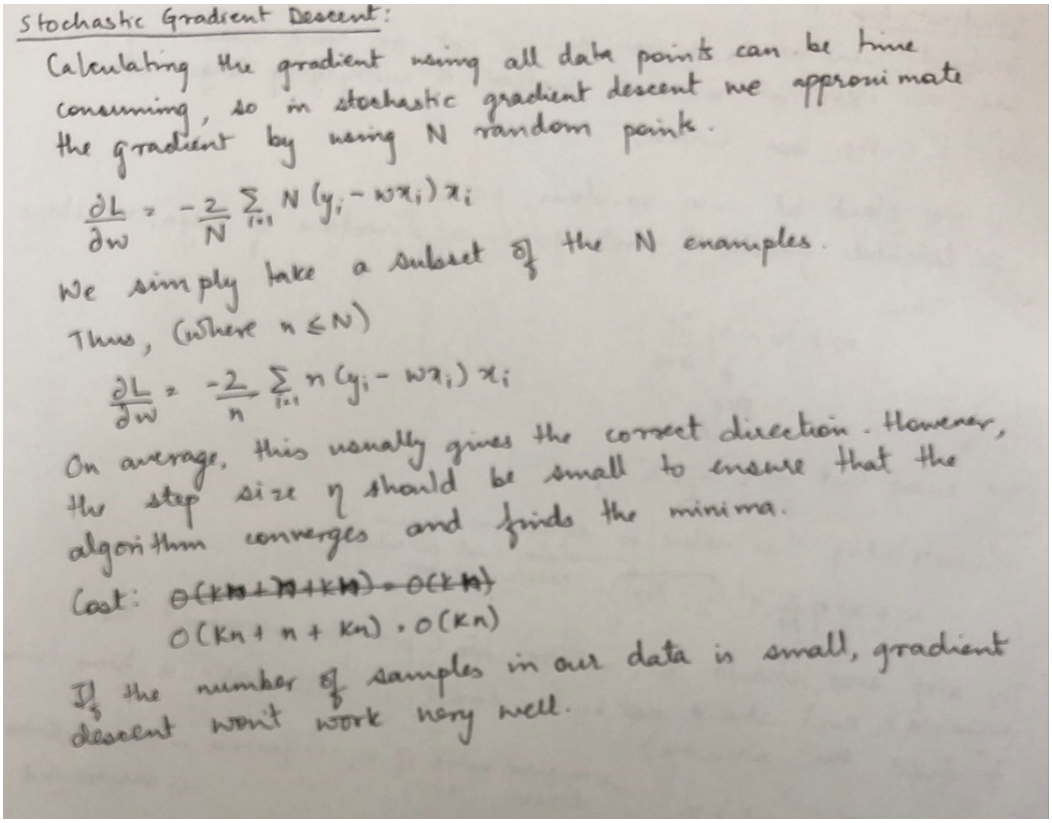


Figure 7: Stochastic Gradient Descent

Notes from tn2381

1 Reproducibility Summary

After learning learning about you should avoided from the last lecture. Here is the summarized list of what you should do:

- Read the literature : because you may not be the first one coming up with the idea. This will help you understand the problem better.
- Formulate your study
- Run a simple pilot
- Analyze the results: which may help you
- Revise your study (null != nil)
- Do a power calculation: mainly to define effect and population size
- Pre-register your plans: declare and post the process above online, for example at AsPredicted. This not only prevents you from fooling yourself and others, but it also help you review your process and work more systematically.
- Run your study
- Create a reproducible report: using makefile and templates created since your pilot study
- Think critically about results
- Disclose everything you did: for example subsetting data, detailed explanation of data collection

1.1 Example of Good Practices in Reproducibility

Prof. Jake gave an example on his project studying the effect of present the result using standard error, standard deviation, and some other techniques on the ability to interpret the true result of readers. From this example, we can learn several good practices which have been implemented and proven to be useful by Prof. Jake himself such as

- Using Mechanical Turk to collect data which is fast and reliable
- Running a pilot test to see the nature of data
- Creating a reproducible analysis framework based on the pilot data
- Creating questions to perform sanity check about participant understanding about the experiment
- Using timestamps data to filter out bots or low quality responses

1.2 Revisiting Standard Deviation and Standard Error

Standard deviation is variation in the population which always remains the same, while standard error can be reduced by sampling more samples.

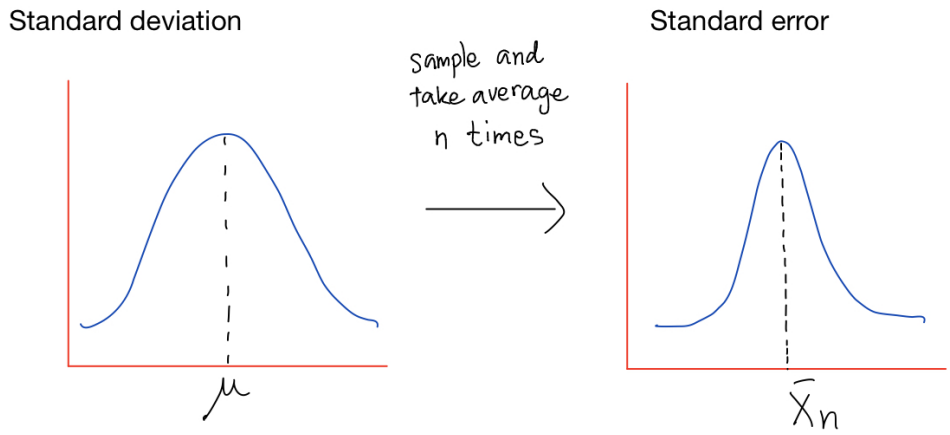


Figure 8: Comparison between standard deviation and standard error

2 Regression

A simple definition of regression is to predict some outcomes from some inputs/ features/ predictors.

$\{y_i\}_{i=1}^N$ or $\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$ are N observations of outcomes (usually scalar values)

$\{X_i\}_{i=1}^N$ or $\begin{bmatrix} X_{11} & \dots & X_{1K} \\ \vdots & & \\ X_{N1} & & X_{NK} \end{bmatrix}$ are N observations of inputs of K dimensions (vector input)

Regression can be simply put as $\hat{y}_i = f(x_i)$, an output of our predictors is equal to our function of inputs.
Goal: We want some functions whose output matches the data well. To quantifying 'well', we introduce

$$\mathcal{L}_i[f] = (y_i - \hat{y}_i)^2$$

called loss function of single data point which is calculated using squared error of actual and predicted outcomes. For the whole data set, we have that

$$\mathcal{L}[f] = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Our goal is to find the function that minimize the loss.

$$f^* = \underset{f}{\operatorname{argmin}}\, \mathcal{L}[f]$$

Motivation: Imagine data is generated by

$$y_i = f(x_i) + \mathcal{E}_i$$

where $\mathcal{E}_i$ is noise/error.

Assumption:

$$\mathcal{E}_i \sim \mathcal{N}(0, \sigma^2)$$

The likelihood of seeing the observed data can be calculated by

$$\begin{aligned} p(\mathcal{E}_i|f) &= p(y_i - f(x_i)|f) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - f(x_i))^2} \\ p(\mathcal{D}|f) &= \prod_{i=1}^{N} p(\mathcal{E}_i|f) \\ &= (\frac{1}{\sqrt{2\pi\sigma^2}})^N \prod_{i=1}^{N} e^{-\frac{1}{2\sigma^2}(y_i - f(x_i))^2} \end{aligned}$$

Under what function f is the probability of the observed data maxizimized?

$$f^* = \underset{f}{\operatorname{argmax}}\, p(\mathcal{D}|f)$$

$p(\mathcal{D}|f)$ is the likelihood, hence f^* is called Maximum Likelihood Solution.

However, dealing with the product term is unpleasant, but thanks the monotonic characteristic of logarithmic function, maximizing $p(\mathcal{D}|f)$ is equivalent to maximizing $log\ p(\mathcal{D}|f)$

$$\begin{aligned} log\ p(\mathcal{D}|f) &= \underbrace{-\frac{N}{2} log\ 2\pi\sigma^2}_{\text{const. w.r.t. } f} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - f(x_i))^2 \\ \underset{f}{\operatorname{argmax}}\, p(\mathcal{D}|f) &= \underset{f}{\operatorname{argmax}}\, log\ p(\mathcal{D}|f) \\ &= \underset{f}{\operatorname{argmax}} -\sum_{i=1}^{N}(y_i - f(x_i))^2 \text{ (removed terms which are constant w.r.t. } f) \\ &= \underset{f}{\operatorname{argmin}} \sum_{i=1}^{N}(y_i - f(x_i))^2 \end{aligned}$$

Hence, we can achieve the Maximum Likelihood Solution by solving Least Squared Error.

How do we search over f?
We can attempt by making another assumption that f is a linear function.

$$\hat{y}_i = f(x_i; w) = w \cdot x = wx_{i1} + ... + wx_{iK}$$

or as a vectorized notation

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{w}$$

We want to find w that minimize the squared error.

$$w^* = \underset{w}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{N} (y_i - w \cdot x_i)^2}_{\mathcal{L}}$$

$$\begin{aligned} 0 = \frac{\partial \mathcal{L}}{\partial w} &= \sum_{i=1}^{N} 2(y_i - wx_i)(-x_i) \\ &= \sum_{i=1}^{N} (y_i - wx_i)(x_i) \\ &= \boldsymbol{X^T(y - Xw)} \\ \boldsymbol{X^T y} &= \boldsymbol{X^T X w} \\ \boldsymbol{w^*} &= \boldsymbol{(X^T X)^{-1} X^T y} \end{aligned}$$

It appears that we can solve for w analytically in a closed form solution. However the complexity of inverting a KxK matrix is $O(K^3)$ and use $O(K^2)$ space making the method not feasible for high dimensional data. Moreover, it is also possible that $\boldsymbol{X^T X}$ is not invertible. Therefore, we need to an iterative algorithm to solve for w.

Gradient Descent: Guess and update

$$\begin{aligned} \text{We update } \boldsymbol{w} &\leftarrow \boldsymbol{w} - \eta \frac{\partial \mathcal{L}}{\partial w} \\ &= \boldsymbol{w} + 2\eta \boldsymbol{X^T(y - Xw)} \end{aligned}$$

This method takes $O(KN)$ time per iteration and $O(KN)$ space.

Stochastic Gradient Descent: Update using w using smaller batch size (or 1 data point)

For stochastic gradient descent, we sample datat to compute gradient. Here is an example of using one data point at a time.

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta(y_i - \boldsymbol{w x_i}) \boldsymbol{x_i}$$

This method only requires $O(mK)$ per iteration where m is the batch size. However, it is more sensitive to the step size, η.

Notes from vs2671

1 Continuation of Reproducibility, Replication Lecture

The Lecture begins with a continued overview into what we should do -

- Read the literature - It is important to know what work has already been done. This might also help you think and understand about the problem more.
- Formulate your study
- Run a simple pilot - a basic version is run to get initial results. This can result in revisions in the study
- Analyze the results
- Revise your study (null != nil)
- Do a power calculation (think about effect size here - reference to homework 2)
- Pre-register your plans
- Run your study
- Create a reproducible report - important for others to check the work
- Think critically about results
- Disclose everything you did - this also includes details around data collection

Next, Professor Jake gives an example of his own research, findings and interpretation. We see an experiment revolving around the probability to win if an item is picked. Multiple tests are run for significance of difference in willingness to pay between the two items.

In order to collect the data, they used Mechanical Turk - a platform that typically allows people to earn small amounts of money for tasks/surveys completed. It was found by him that the average responses by a random set of people on Turk were not typically different from how someone who knew about the subject would respond.

We also see a brief overview of the data file they get back. Typically, a lot of data is collected and can be used for different purposes -
eg. We revisit the boulder experiment covered two lectures before. Data like when they saw start boulder, timestamps so that they can check for bots and filter them out

The steps he has followed closely match the description of what we should do stated above. They can be briefly summarized as -

Data collection $\rightarrow$ Run pilot $\rightarrow$ See Rmd and finalize result $\rightarrow$ Study to check $\rightarrow$ Analysis and summary of findings

1.1 Brief revisit to standard error and deviation

For a sample size of 10 people, we compute the average value for a quantity

Standard error - variation in the average decreases with more samples

Standard deviation - Measure of uncertainity in the population

$$\sigma = \frac{\sigma_x}{N^{1/2}}$$

The standard deviation graph is relevant for effect size. If means are different but the variation overlaps a lot, then that is not very significant.

2 Regression

Regression analysis basically refers to using the available data to infer relationships from variables and predict future outcomes.

2.1 Goals

The primary goals of regression analysis are to -

- Predict future outcomes
- Explain associations between predictors and outcomes
- Describe or summarize outcomes under different conditions

An example of regression analysis would be SAT scores for Asian vs Hispanic students.

2.2 Prediction problem

We basically want to find a function f whose output matches the data well such that

$$y_i = f(x_i)$$

Here, y is the output/outcome vector represented by

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

And x is the input vector such that there are n training samples of d dimensions/features each

$$\begin{bmatrix} x_{11} & \dots & x_{1d} \\ \vdots & & \\ x_{n1} & & x_{nd} \end{bmatrix}$$

2.3 Loss Function

To arrive at our desired function mapping f(x), we define a loss function. Our goal is to **minimize this loss function**

A reasonable choice of loss function is Least squares error

$$L_i[f] = \frac{1}{N}\sum_{1}^{N}(y_i - f(x_i))^2$$

Another choice is the absolute loss function

$$L_i[f] = \frac{1}{N}\sum_{1}^{N}|y_i - f(x_i)|$$

Here y_i is actual outcome and $f(x_i)$ is the predicted outcome.

2.4 Maximum Likelihood Interpretation

Assumption: We imagine that the data is generated by

$$y_i = f(x_i) + \epsilon_i$$

such that $\epsilon_i \sim N(0, \sigma^2)$
We can then calculate the likelihood of seeing the observed data D as -

$$p(D|f) = \prod_{i=1}^{N} p(\epsilon_i|f) = (\frac{1}{\sqrt{2\pi\sigma^2}})^N \prod_{i=1}^{N} e^{-\frac{1}{2\sigma^2}(y_i - f(x_i))^2}$$

Observe : We have assumed independence here.
What we basically want to find is under what function f is the probability of the observed data maximized.

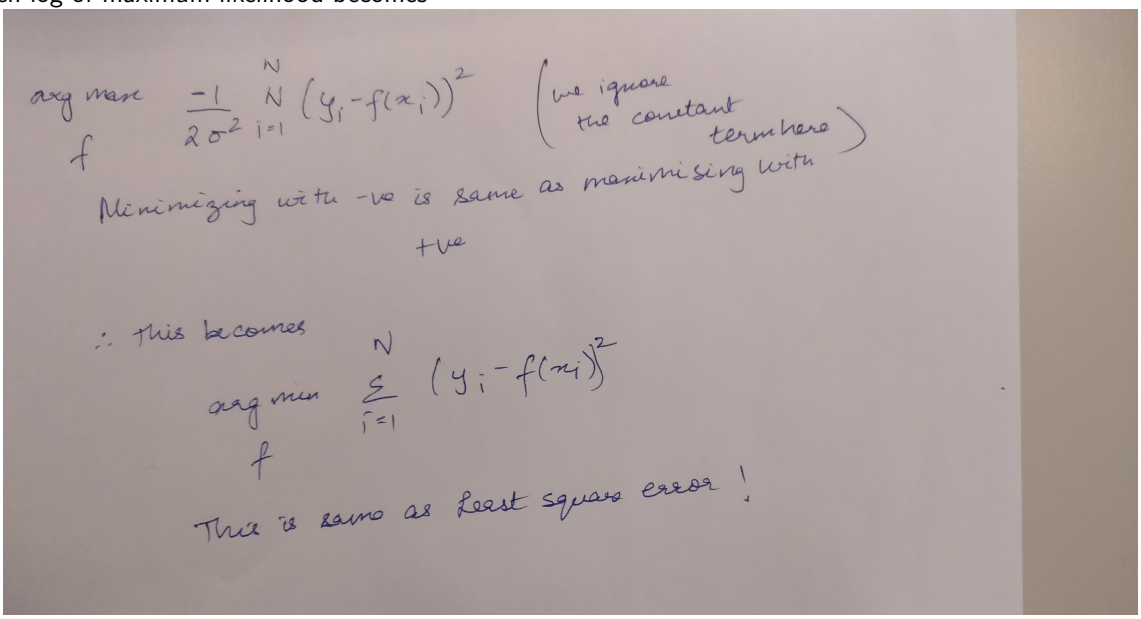
This is given by

$$f^* = \underset{f}{\operatorname{argmax}}\, p(D|f)$$

where f^* Maximum Likelihood Solution.

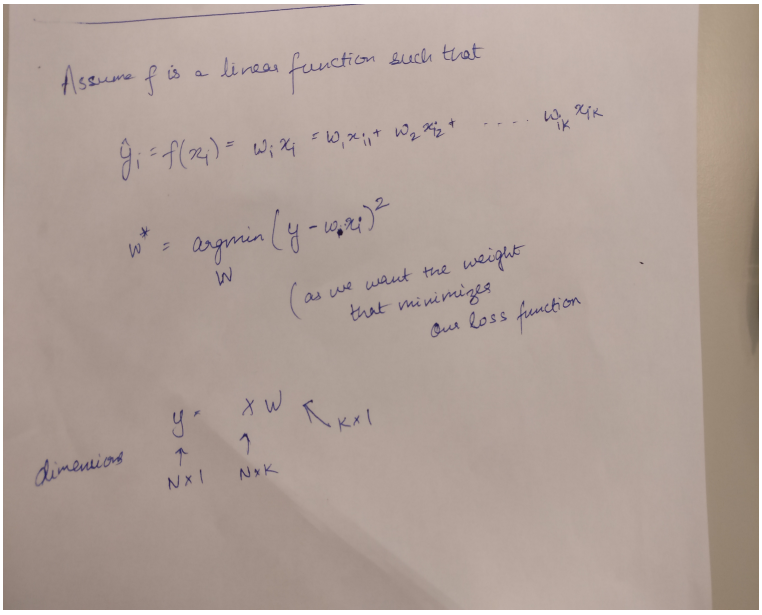
2.5 Maximum Log Likelihood and relation to Least square error

Since it is difficult to calculate in the product form stated above, we take a log to deal with the products! It is okay to do this because of the monotonically increasing nature of log function.
Then log of maximum likelihood becomes -

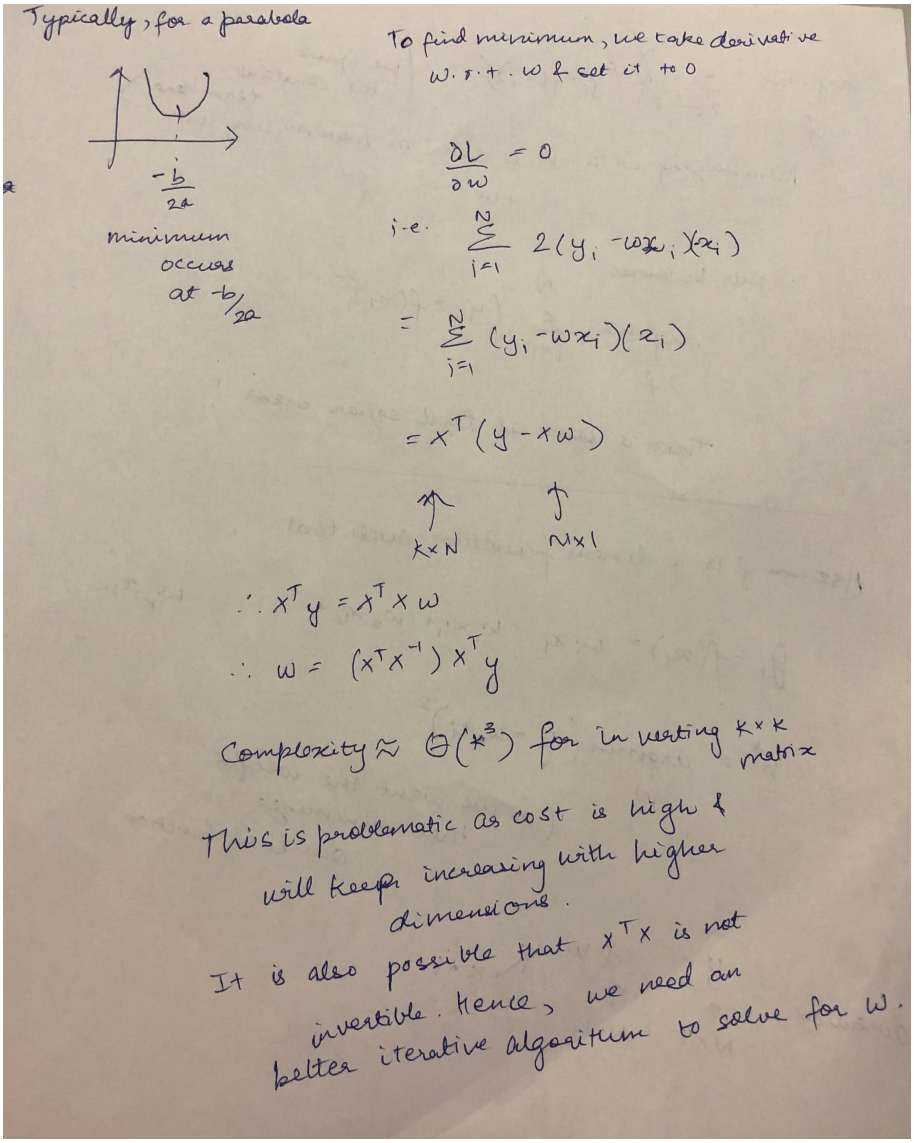


Hence, we can achieve the Maximum Likelihood Solution by solving Least Squared Error.

2.6 Computing Desired weight vector



Continued computation given on next page



2.7 Gradient Descent

We update weights as

$$w := w - \alpha \frac{\partial L}{\partial w}$$

where L is the loss function
which is

$$w := w - \alpha \frac{2}{N} X^T (y - Xw)$$

as the update rule at each step.

where α is called the learning rate, which is the step size we take at each iteration in the direction of negative gradient. This method take $O(kN)$ to compute the update for w at each step.

There are variations to this vanilla form of gradient descent- One such variation is the stochastic gradient descent where we update w using a smaller batch size (or 1 data point)

$$w := w - \alpha(y_i - w \cdot x_i)x_i$$

This method only requires $O(mK)$ where m is the batch size